NUMERICAL SIMULATION OF THE PIVOTAL INTERNAL COMBUSTION ENGINE

A thesis submitted in partial fulfilment of the

requirements for the degree

of

Masters of Mechanical Engineering

in the

University of Canterbury

by

A. C. Saville

University of Canterbury

2002

ACKNOWLEDGMENTS

I would especially like to thank Dr Ian Huntsman who has provided outstanding academic supervision for the project, Paul McLachlan and Nick Turner from Pivotal Engineering for their continued support and encouragement as the project developed, and Mr Ted Mace, of Mace Engineering, who has supported the development of the Pivotal engine and the simulation project. I would also like to express appreciation to all the staff from the Department of Mechanical Engineering who have been extremely helpful and friendly in assisting the research project. In particular I would like to thank Mr Graeme Harris from the aeronautical laboratory. This project has been funded by Technology New Zealand Foundation for Science and Technology, under graduate research in industry fellowship (GRIF) contract number PVEX9901. Finally I must thank my partner Joanne Brokenshire for her tremendous support and encouragement.

Allan Saville

ABSTRACT

This thesis describes the development of a one-dimensional internal combustion engine simulation program for the Pivotal two-stroke engine. The Pivotal two-stroke engine has many features in common with the standard reciprocating piston engine, but differs in its novel kinematics, which are based around a four bar linkage. The new engine arrangement opens up many new design options and required a flexible and specific simulation tool for research and development.

The initial project goals were to develop a simulation code, validate the code against engine data and develop a user interface for easier application. However, it was not possible to realise all of these goals and the project was mostly concerned with the development of the simulation numerical code. Validation will be required before the simulation tool can be used with confidence. Engine simulation is a relatively mature field of research and the simulation program includes many established standard methods.

The simulation incorporates a standard single zone thermodynamic cylinder model and a Riemann quasi-one-dimensional finite volume gas dynamics scheme, which includes total variation diminishing variable extrapolation. The calculation of flow through valves uses standard equations for orifice flow and the application to pipe boundary conditions utilises the propagation of characteristic information out of the pipe boundary. Simulation of reed valve deflection has been implemented with the finite element method using direct integration in time.

The entire code has been implemented in the Fortran programming language using Compaq Visual Fortran 6.5. To establish a flexible simulation the program has been devised so that the input text data file determines the arrangement of the engine, allowing most engine configurations to be modelled with the one simulation tool. Summary data, including engine power, is outputted to a text file, which can be read by any text editor. Detailed simulation results are written to Matlab mat-files, which require Matlab 5.3, or greater, to open and analyze.

In its current form the simulation code is capable of running arbitrary engine arrangements, including the Pivotal engine, and can predict the performance of these engines. However, the code is un-validated against real engine data. Where possible sub-models have been tested and proven against special test cases that have an analytic solution. The next logical extension beyond the current project is to thoroughly validate and compare the engine simulation against real engine test data. Other areas in which the code could be improved include; extending the simulation code to incorporate more advanced models and improving the ease with which engine simulations can be set-up, run and analysed.

LIST OF FIGURES

TABLE OF CONTENTS

INTRODUCTION

Pivotal engine

The Pivotal engine, shown in Figure 1-1, is a new two-stroke internal combustion (IC) engine that has a similar mode of operation as traditional two-stroke engines but has modified kinematics due to a four-bar linkage arrangement. This compares to the traditional reciprocating piston internal combustion engine that utilises a slider crank mechanism. In a Pivotal engine the conventional piston is replaced by a rectangular pivot piston, which rotates about a pivot bearing on the engine block. The increased complexity incurred by using the four bar linkage is offset by gains in the kinematic restraint of the piston, and the ability to have asymmetric compression and expansion strokes.

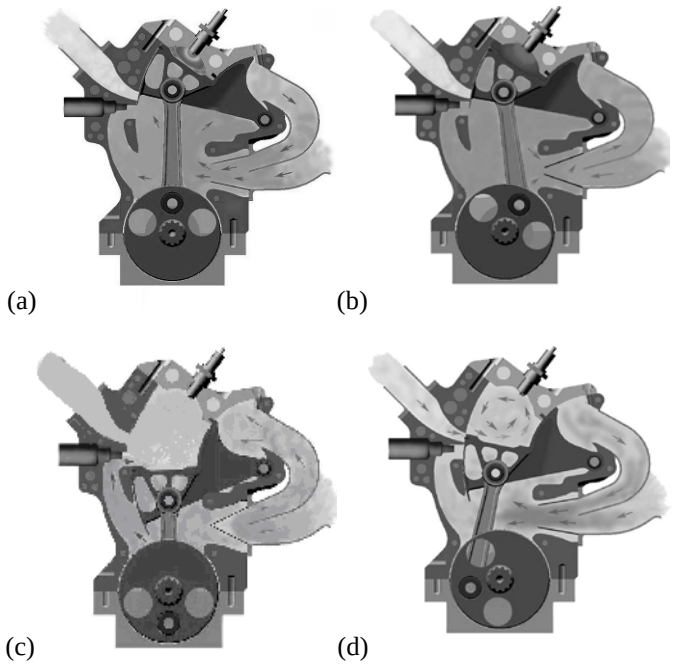


Figure 1-1, (a). TDC, Ignition has occurred. Crankcase filling with fresh charge through reed valve. (b). Expansion power stroke. Reed valve is closing. (c). BDC, Exhaust gases expelled and fresh charge is filling the combustion chamber. Fuel metered by transfer port fuel injectors. (d). Beginning of compression stroke, export port closing.

In a slider-crank engine the piston skirt and seals, perform the dual role of reacting side thrust loads on the piston and sealing the cylinder. The Pivotal engine however

separates these two tasks. Figure 1-2 shows a schematic diagram of the Pivotal pivot piston.

Kinematic restraint of the pivot piston is achieved by the use of a pivot bearing and the connecting rod small end. The skirts and seals then only need to have oil control and combustion chamber sealing functions. This allows greater opportunity for tuning each part for their dedicated function in the engine.

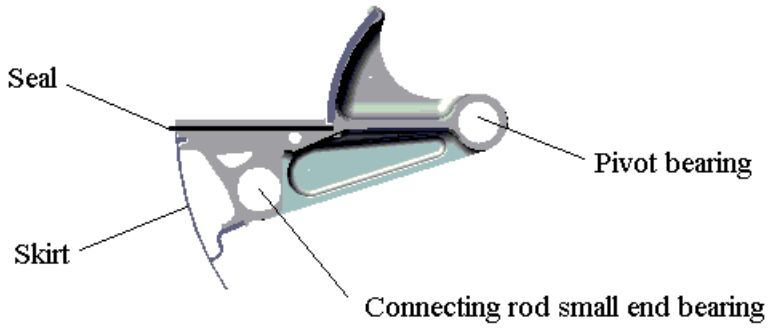


Figure 1-2. Schematic diagram of the Pivotal pivot piston.

The extra degree of freedom in the pivot piston / crankcase linkage, i.e. the four-bar linkage, also increases the possibilities for optimising the cylinder volume as a function of crank angle with the purpose of maximum thermodynamic efficiency and cylinder scavenge. Due to the extra kinematic link, the volume crank angle relationship can also be made asymmetric in the compression and expansion stroke allowing asymmetric port open timing.

The Pivotal engine permits water circulation to the pivot piston through the pivot bearing. This has lead to a simple and trouble free implementation of water-cooled piston technology, tested at engine speeds up to 6000 rpm.

Project goals

The project goal was to develop an advanced two-stroke engine simulation package for Pivotal Engineering ltd that are developing a two-stroke Pivotal engine. Three main areas of the project were established.

(1). Development of a core physical model simulation code.

(2). Engine testing and validation of the code.

(3). Development of post and pre processing user interface.

From the onset of the project the intention was to complete all three tasks however it became apparent that insufficient resource was available. The revised project goal became developing a comprehensive simulation code with simple text file inputs and carrying out limited validation with real engine data to verify the predictions of the engine simulation. Commercial simulation packages are available for conventional reciprocating internal combustion engines, however due to the unique arrangement of the Pivotal engine these packages were not entirely suitable.

Background

The objective of engine simulation is to develop a predictive numerical model of the physics occurring within an engine. Not all processes can be predicted based on an analysis of physical considerations and empirical model are introduced. As a result engine simulation uses a combination of predictive and empirical methods.

Simulation facilitates investigation of a broad range of engine concepts in a condensed time without building any prototypes. The detailed information about engine operation available, also gives valuable insights and understanding of the physical processes occurring in an engine. Simulation allows rapid engine development, a broad scope of investigation, and provides in depth understanding of engine processes. As a result accurate simulation leads to reduced research and development costs.

The two main predictive models are a quasi-one-dimensional computational fluid dynamics (CFD) model for pipe flow and a thermodynamic control volume analysis of cylinder processes. A boundary condition model calculates the flow through a valve between a cylinder and pipe, providing the boundary condition for the pipe and cylinder. In the current simulation a further finite element model has been incorporated to model dynamic deflection of reed valves in two stroke engines. Empirical models are generally introduced to estimate friction and heat transfer and are invariably required to model combustion.

Numerical discretisation of the engine system results in a set of differential equations that describe the complete operation of an engine as a function of time. These equations are integrated in time using an explicit time stepping scheme. The simulation is then

repeatedly integrated through the engine cycle until convergence between one engine cycle and the next is achieved.

Gas dynamics

Pioneering research in the simulation of manifold pipe flow, using digital computers, invariably used the method of Characteristics (MOC). The text by Benson [1] gives a thorough introduction of the MOC. The MOC gives valuable insights into the nature of flow problems in pipes, particularly at boundaries. For this reason the MOC remains an important method, however it has largely been superseded by the finite volume method (FVM). Blair [2] has also introduced a numerical scheme base on the propagation of finite amplitude waves in pipes.

The MOC numerical scheme is limited to first order accuracy and does not conserve mass, energy and momentum in the flow field. The MOC also has difficulty accurately resolving flow discontinuities. The finite volume method (FVM) has addressed these shortcomings and has contributed to improved accuracy in engine manifold simulations. Pearson [3] and the text by Winterbone and Pearson [4], give a detailed overview and comparison of current quasi-one-dimensional computation fluid dynamics schemes currently used in engine simulation. Of these a standard total variation diminishing (TVD) Godunov scheme was selected for the current simulation project. Toro [5] and Hirsch [6] give detailed theory of this method.

The Godunov scheme uses the solution to the Riemann problem, a generalised form of the shock tube problem, to evaluate the inter-cell flux at the interface of the pipe discretisation. In multi-dimension CFD an approximate solution of the Riemann problem is normally sort, due to the numerical cost of evaluating the solution. As the number of volumes in quasi-one-dimensional gas flow is relatively, very low, an exact solution was used in the engine simulation. The Godunov scheme has the advantage of including the characteristic information in the calculation of the inter-cell flux thus correctly modelling the physics of propagation in pipe flow.

Linear piecewise reconstruction of the volume data extended the scheme to second order spatial accuracy. This needed a TVD slope limiter to avoid oscillations about discontinuities in the flow.

Thermodynamics

Thermodynamic modelling of cylinders and other engine volumes rely on the application of the first law of thermodynamics, 'conservation of energy'. Benson [1] and Heywood [7] describe the application of this law in engine simulation. The type of model resulting from this approach can be either single or multi zone. For single zone models, combustion is considered as simple heat release, whereas multi zone models calculate separate zones for the burnt and unburnt gases in the cylinder. The method adopted here was a single zone cylinder with a Wiebe heat release function.

To account for the variation of chemical species during combustion, an equilibrium chemistry calculation, due to Olikara and Borman [8], was incorporated into the model. The time scales of the chemical reactions in the equilibrium model are significantly shorter than the time scale of the combustion event therefore the model adequately predicts the concentrations of the main chemical species formed during combustion. The formation of NO_x pollutants however is rate dependent within the combustion time scale and thus requires an additional kinetic chemistry model to predict pollutants. As the multi-zone models better predict the temperature during combustion, kinetic chemistry is best incorporated within a two-zone simulation. As the current simulation code is only single zone, NO_x formation calculations were omitted. The variation of heat capacity and internal energy of each of the species was calculated from a polynomial as a function of temperature. The coefficients for these polynomials where taken from the NIST [9] chemistry webbook.

To calculate heat transfer from the cylinder the empirical Annaud heat transfer model, as described in Heywood [7], was incorporated into the cylinder simulation. The Annaud model treats the cylinder as a short pipe and uses standard non-dimensional analysis to establish an empirical method that predicts the mean instantaneous heat transfer coefficient for the cylinder. In the case of the Pivotal engine an effective cylinder diameter was assumed, and the heat transfer coefficient calculated as per the Annaud method for standard engines.

Valve pipe boundary model

Valve elements conceptually act at a point connecting a pipe to other parts of the engine. The flow is assumed to be quasi-steady state at the boundary. The correct modelling of the physics at the pipe boundary has an important effect on the overall

accuracy of the engine model. Benson [1] introduced pipe boundary condition treatments that are widely accepted, and the work by Winterbone and Pearson [2] presents Benson's work, in the light of up to date research. Vandevoorde Et al [10] also present a similar method of imposing boundary conditions.

To remain well posed the implementation of boundary conditions needs to remain consistent with the way in which information propagates in gas dynamics. Hirsch [4] shows that the imposed boundary conditions need to satisfy the compatibility equations or Riemann invariants for the outgoing characteristics. The energy equation, for flow between the valve and a reservoir, with appropriate assumptions, then closes the set of equations needed to update the boundary condition. In the engine simulation code developed here, these standard techniques have been used to establish a model for valve elements at pipe / cylinder boundaries.

Reed valve deflection

Simulation of reed valve deflection has been tackled with a variety of techniques by various researchers. The most simple is to model the reed as a discrete spring-mass-damping system. Whilst adequate in simple simulations, it does not model the effect of higher order modes of vibration on the response of the reed. The next level of sophistication is to directly solve the equations of motion for a beam, or plate, with assumed mode shapes for the vibration of the reed. Fleck and Blair [11], Fleck, Blair and Houston [12], and the more recent work by Royo and Perez [13], detail this approach. The method gives excellent results. However all possible mode shapes, including those when the reed is in partial contact with the lift stop and reed block, need to be assumed before hand to fully capture the reed dynamics.

The method adopted in the current simulation, was to form a standard finite element model of the reed, using beam elements, and directly integrate the reed deflection in time. The texts by Astley [14], and Zeinkiewicz [15] detail this method. The direct integration approach has the advantage that no prior knowledge of the reed modes of vibration is necessarily required. Although reed bounce was not considered, the method can be extended to include it. The FEM can also be extended to two dimensional plate elements, which would be a natural extension for future development.

Complete engine simulation

Each of the engine elements (valves, cylinders, pipes etc), except the reed valve, were incorporated into a global four stage Runge-Kutta time stepping algorithm achieving fourth order accuracy in time for the global simulation. The finite element reed valve deflection model was integrated separately using the implicit Newmark time integration method.

To establish a flexible simulation tool, the program has been devised so that the input file determines the arrangement of the engine. With this arrangement almost any engine configuration may be modelled. Each simulation element is modular so that they may connect to other elements at run time.

The engine simulation program provides a design tool allowing analysis of a broad range of concepts and identification of the best engine arrangements to further investigate with real prototypes. Simulation also provides valuable insights into the physical processes occurring within the engine, and allows a far broader scope of investigation than with physical prototypes alone.

References

[1] Benson, R. S. (1982) The Thermodynamics and Gas Dynamics of Internal-Combustion engines, Clarendon press, Oxford. Vol 1 and 2.

[2] Blair, G. P. (1996) Design and simulation of two-stroke engines, SAE, Warrendale.

[3] Pearson, R. J. (1994) Numerical methods for simulating gas dynamics in engine manifolds, PhD. Thesis, Department of mechanical engineering, UMIST, Manchester, UK.

[4] Winterbone, D. E. and Pearson. R. J. (2000) Theory of engine manifold design-Wave action methods for IC engines, SAE, Warrendale.

[5] Toro, E. F. (1997) Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer, New York.

[6] Hirsch. C. (1990) Numerical computation of internal and external flows, John Wiley & Sons, England, Vol 2, chapter 19.

[7] Heywood, J. B. (1998) Internal combustion engine fundamentals, McGraw-Hill, New York.

[8] Olikara, C. and Borman, G. L. (1975) A computer program for calculating properties of equilibrium combustion products with some applications to I.C engines, SAE paper 750468.

[9] NIST (2001) Chemistry WebBook, http://webbook.nist.gov/chemistry/

[10] Vandevoorde. M., Vierendeels. E., Dick. E. and Sierens. R. (1998) A new total variation diminishing scheme for the calculation of the one-dimensional flow in inlet and exhaust pipes of internal combustion engines, Proc Instn Mech Engrs, Vol 212 part D, Pg 437-448

[11] Hinds. E. T. and Blair. G. P. (1978) Unsteady Gas Flow Through Reed Valve Induction Systems, SAE paper 780766.

[12] Fleck. R., Blair. G. P. and Houston. R. A. (1987) An Improved Model for Predicting Reed Valve Behaviour in Two-Stroke Cycle Engines, SAE paper 871654.

[13] Royo, R. and Perez, A. (1995) Modelling of reed valves. Application to 2-S internal combustion engines, ICE, v23, Proceedings of the ASME Internal Combustion Engine Division Spring Meeting .

[14] Astley, R. J. (1992) Finite elements in solids and structures, Chapman and Hall, London.

[15] Zienkiewicz, O. C. and Taylor, R. L. (1990) The Finite Element Method, 4th edition, McGraw

GAS DYNAMICS

The tuning of the exhaust and inlet ducts of an internal combustion (IC) engine is important because it has a significant effect on the gas exchange process and overall performance of the engine. Any engine simulation must model the dynamics of these elements if acceptable accuracy is to be achieved.

For this simulation program a quasi-one-dimensional explicit MUSCL finite volume gas dynamics scheme was used. The acronym MUSCL standing for Monotonic Upstream-centred Scheme for Conservation Laws. In the MUSCL approach variables are reconstructed piecewise linear over each volume and the fluxes are calculated using the solution to the Riemann problem evaluated at the volume interfaces. The current implementation used an exact Riemann solver, due to Toro[1]. To avoid spurious oscillations total variation diminishing (TVD) slope limiting was introduced with a Van Leer slope limiter.

The advantage of using the Riemann problem to evaluate fluxes is that the upwind propagation of characteristic information is contained within the calculation of the Riemann problem at each volume interface, providing a physically rigorous method.

Governing Equations

The integral-partial differential equations, which describe compressible quasi-one-dimensional unsteady flow for a control volume V, are given by.

$$\frac{\partial}{\partial t}\iiint W dV + \iint F dA + G = 0 \tag{2.1}$$

where

$$w = \begin{bmatrix} \rho \\ \rho u \\ \rho\left(c_v T + \frac{1}{2}u^2\right) \end{bmatrix} \tag{2.2}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u\left(c_v T + \frac{1}{2}u^2\right) + pu \end{bmatrix} \tag{2.3}$$

$$G = \begin{bmatrix} 0 \\ \mathrm{f} - \int p \dfrac{\partial A}{\partial x} dx \\ \mathrm{q} \end{bmatrix} \tag{2.4}$$

where

ρ = density, u = velocity, T = temperature, p = pressure, c_v = specific heat capacity at constant volume, A = cross section area, x = distance along pipe,

$$\mathrm{f} = A_s f \tfrac{1}{2} \rho u^2 \tag{2.5}$$

f = friction force, f = pipe friction factor, A_s = surface area

$$\mathrm{q} = hA_s(T - T_{wall}) \tag{2.6}$$

q = heat transfer, h = heat transfer coefficient, T = gas temperature, T_{wall} = pipe wall temperature

The flow is assumed to be an ideal gas so that the ideal gas law can be used to close the set of equations.

Numerical discretisation

Equation (2.1) is discretised by the finite volume method. Figure 2-1 shows a section of pipe divided into discrete volumes.

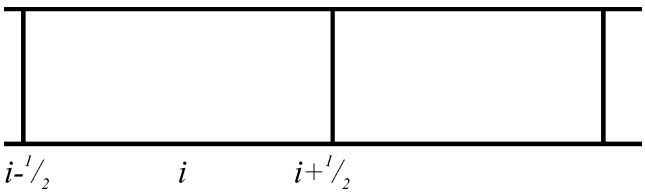


Figure 2-1. Discrete pipe volume.

The discrete form of Equation (2.1) for the pipe volume is given by:

$$\Delta W_i = \frac{\Delta t}{V}\left[F_{i+\frac{1}{2}} A_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} A_{i-\frac{1}{2}} + G_i\right] \tag{2.7}$$

Here V is volume, W represents the averaged conserved variable for the volume, and the fluxes, F and G, are time averaged values for the time step Δt.

Flux evaluation

The MUSCL method approximates the inter-volume fluxes by evaluating the Riemann problem at each volume interface. The basic assumption of the method is that at any time step the variables have a piecewise distribution across each volume with a discontinuity at each volume interface, as shown in Figure 2-2. A local Riemann problem is defined at the interface and solved to give the inter-volume flux. The method incorporates the fundamental physics of the hyperbolic conservation laws in the flux calculation. A complete derivation of the exact Riemann solution can by found in the text by Toro [1], but has only been outlined here.

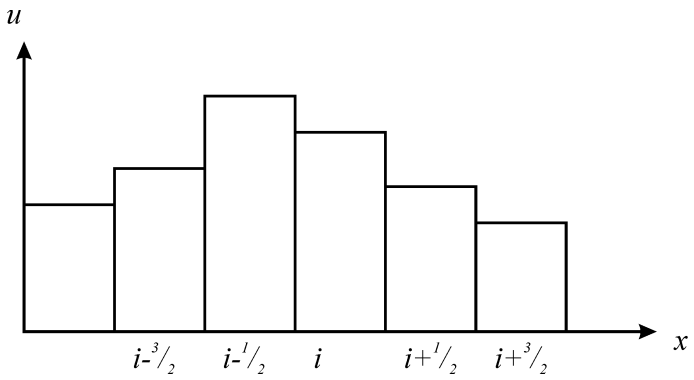


Figure 2-2. Piece wise constant representation of pipe data.

The Riemann problem is a more general form of the shock tube problem that permits discontinuity in velocity as well as pressure and density. The shock tube problem is physically realisable with a gas filled pipe divided into two sections by a diaphragm. On each side of the diaphragm the pressure and density initial conditions are different. The shock tube problem is initiated by rupturing the diaphragm and allowing the resulting wave pattern to develop. The waves resulting from the Riemann problem form in one of four patterns as shown in Figure 2-3. The initial discontinuity between the left and right data sates is located at $x = 0$ and the figure illustrates the progress of the resulting wave pattern in time. Depending on the left and right initial conditions the left and right waves can be any combination of shocks and rarefactions, always with a contact discontinuity in between.

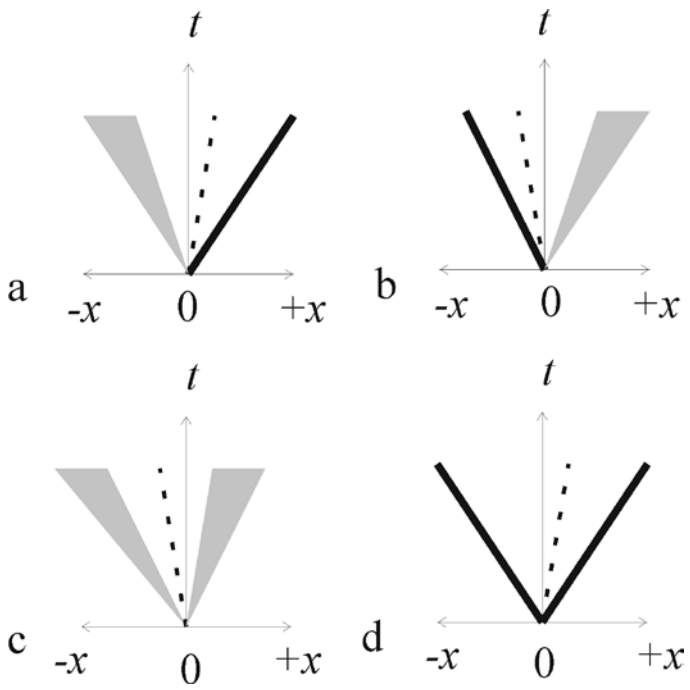


Figure 2-3. Possible wave patterns from the Riemann problem. (a) Left rarefaction, centre contact discontinuity, right shock. (b) Left shock, centre contact discontinuity, right rarefaction. (c) Left and right rarefaction, centre contact discontinuity. (d) Left and right shock, centre contact discontinuity.

Form of the Riemann solution

The region between the left and right moving waves in the Riemann problem is known as the star region, as labelled in Figure 2-4. Pressure and velocity are constant in the star region but density is discontinuous across the contact discontinuity. The solution to the Reimann problem is formulated by developing an expression for the pressure in the star region. The expression does not have a closed form but can be solved for pressure by the Newton-Raphson or similar iterative method. Once the pressure in the star region has been found the other variables are easily determined.

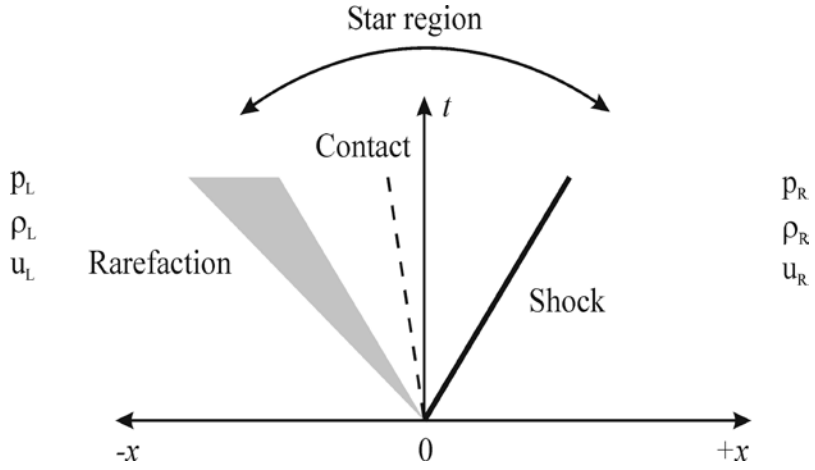


Figure 2-4. Star region of the Riemann problem.

The pressure in the star region, denoted p_*, of the Riemann problem is found by solving the following equation.

$$f_L(p_*) + f_R(p_*) + u_R - u_L = 0 \tag{2.8}$$

The derivation of Equation (2.8) can be found in Toro [1]. The method involves developing two expressions for the velocity behind the left and right wave in the star region as a function of p_* and the left and right initial data states respectively. As u_*, the velocity in the star region, is constant, the difference of the velocity, calculated separately for behind the left and right waves, must be zero. Applying this reasoning leads to Equation (2.8). The pressure Equation (2.8) does not have a closed form but was solved by the Newton-Raphson iterative technique. The form of the functions, f_L, f_R, are determined by considering the Riemann invariant across a rarefaction, or the Rankine-Hugoniot across a shock. Once p_* has been found u_* and the solution to the Riemann problem at any point *P(x,t)* follow.

Sampling the Riemann problem

Once the pressure and velocity in the star region have been calculated the solution at any point, *P(x,t)*, can be easily deduced. The position of the wave pattern relative to *P(x,t)* is determined by considering the speed of each wave. The relations given below are then applied to determine the state at *P(x,t)*.

SHOCK

We can solve the Riemann problem for p_* and u_* and application of the Rankine-Hugoniot conditions leads to an expression for the density in the region between the shock and the contact discontinuity, and the velocity of the left moving shock.

$$\rho = \rho_L \left(\frac{\frac{p_*}{p_L} + \frac{\gamma - 1}{\gamma + 1}}{\left(\frac{\gamma - 1}{\gamma + 1} \right) \frac{p_*}{p_L} + 1} \right) \tag{2.9}$$

$$S_{shock} = u_L - c_L \left[\frac{\gamma + 1}{2\gamma} \frac{p_*}{p_L} + \frac{\gamma - 1}{2\gamma} \right]^{\frac{1}{2}} \tag{2.10}$$

Similarly for a right moving shock.

$$\rho = \rho_R \left(\frac{\frac{p_*}{p_R} + \frac{\gamma - 1}{\gamma + 1}}{\left(\frac{\gamma - 1}{\gamma + 1} \right) \frac{p_*}{p_R} + 1} \right) \tag{2.11}$$

$$S_{shock} = u_R + c_R \left[\frac{\gamma + 1}{2\gamma} \frac{p_*}{p_R} + \frac{\gamma - 1}{2\gamma} \right]^{\frac{1}{2}} \tag{2.12}$$

RAREFACTION

The variation of the state variables within a left moving rarefaction wave is calculated by considering the characteristic through the origin to any point within the rarefaction, and the Riemann invariant which remains constant across the rarefaction.

Characteristic:

$$\frac{dx}{dt} = \frac{x}{t} = u - c \tag{2.13}$$

Riemann invariant:

$$u_L + \frac{2c_L}{\gamma - 1} = u + \frac{2c}{\gamma - 1} \tag{2.14}$$

Using Equations (2.13) and (2.14), and assuming isentropic flow in the expansion wave leads to the following expressions for the state within the rarefaction:

$$\rho\,(x) = \rho_L \left(\frac{2}{\gamma + 1} + \frac{\gamma - 1}{c_L(\gamma + 1)} \left(u_L - \frac{x}{t} \right) \right)^{\left(\frac{2}{\gamma - 1} \right)}$$

$$u(x) = \frac{2}{\gamma + 1} \left(c_L + \frac{\gamma - 1}{2} u_L + \frac{x}{t} \right) \tag{2.15}$$

$$p(x) = p_L \left(\frac{2}{\gamma + 1} + \frac{\gamma - 1}{c_L(\gamma + 1)} \left(u_L - \frac{x}{t} \right) \right)^{\left(\frac{2}{\gamma - 1} \right)}$$

The speed of the head and tail of the left moving rarefaction are given by:

$$S_{head} = u_L - c_L$$

$$S_{tail} = u_* - c_* \tag{2.16}$$

u_* and p_* are known and c_* is found by assuming isentropic conditions across the rarefaction.

Similarly for a right moving rarefaction the state within the expansion is given by

$$\rho(x) = \rho_R\left(\frac{2}{\gamma+1} - \frac{\gamma-1}{c_R(\gamma+1)}\left(u_R - \frac{x}{t}\right)\right)^{\left(\frac{2}{\gamma-1}\right)}$$

$$u(x) = \frac{2}{\gamma+1}\left(-c_R + \frac{\gamma-1}{2}u_R + \frac{x}{t}\right) \tag{2.17}$$

$$p(x) = p_R\left(\frac{2}{\gamma+1} - \frac{\gamma-1}{c_R(\gamma+1)}\left(u_R - \frac{x}{t}\right)\right)^{\left(\frac{2}{\gamma-1}\right)}$$

The speed of the head and tail of the right moving rarefaction are given by

$$S_{head} = u_L + c_L$$

$$S_{tail} = u_* + c_* \tag{2.18}$$

Variable extrapolation

Piecewise constant representation of volume data in the pipe discretisation is equivalent to a first order spatial discretisation. Using linear piecewise representation yields second order spatial accuracy. Figure 2-5 shows a piecewise linear reconstruction of the pipe variables in each volume of the pipe discretisation.

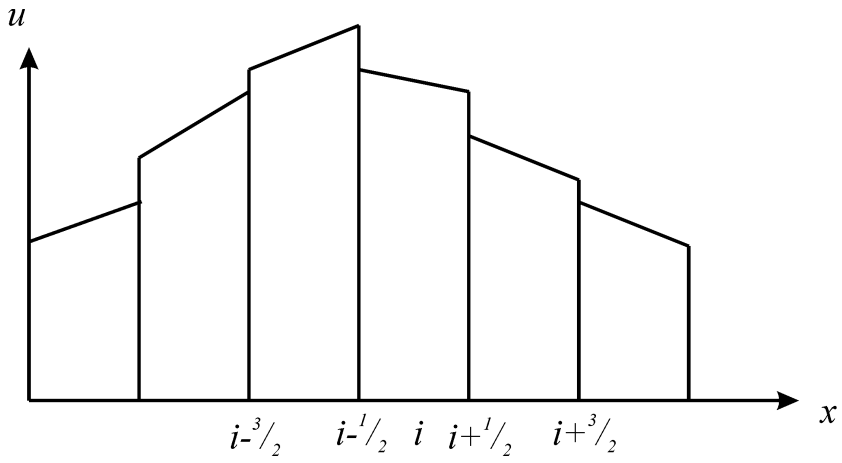


Figure 2-5. Piecewise linear extrapolation of pipe volume data.

The order of accuracy obtained with linear piecewise data reconstruction can be illustrated by considering a truncated expansion for the cell averaged variables [2].

$$u(x) = u_i + \frac{1}{\Delta x}(x - x_i)\partial u_i + \frac{3\kappa}{2\Delta x^2}\left[(x - x_i) - \frac{\Delta x^2}{12}\right]\partial^2 u_i \tag{2.19}$$

The average value is defined by.

$$u_i = \frac{1}{\Delta x}\int_{i-1/2}^{i+1/2} u(x)dx \tag{2.20}$$

When $\kappa = {}^1/_3$ equation (2.19) becomes the truncated Taylor series for *u(x).*

Using central differences to approximate the slopes in each volume.

$$\partial u_i = \frac{u_{i+1} - u_{i-1}}{2} \tag{2.21}$$

$$\partial^2 u_i = u_{i+1} - 2u_i + u_{i-1} \tag{2.22}$$

In the Riemann problem, only the values at volume boundaries are required therefore substituting,

$$x = x_i \pm \frac{\Delta x}{2} \tag{2.23}$$

and equations (2.21)-(2.22), into (2.19), gives the left and right data states at each boundary.

$$u^L_{i+½} = u_i + \tfrac{1}{4}(1-\kappa)(u_i - u_{i-1}) + \tfrac{1}{4}(1+\kappa)(u_{i+1} - u_i) \tag{2.24}$$

$$u^R_{i-½} = u_i - \tfrac{1}{4}(1+\kappa)(u_i - u_{i-1}) - \tfrac{1}{4}(1-\kappa)(u_{i+1} - u_i) \tag{2.25}$$

By the introduction of the parameter κ in equation (2.19) the nature of the extrapolation can be altered. Only for κ = $^1/_3$ is the extrapolation parabolic. For other values the extrapolation is considered linear with truncation errors.

κ = -1, corresponds to a one sided extrapolation of the left and right data states.

$$u^L_{i+½} = u_i + \tfrac{1}{2}(u_i - u_{i-1}) \tag{2.26}$$

$$u^R_{i-½} = u_{i+1} - \tfrac{1}{2}(u_{i+2} - u_{i+1}) \tag{2.27}$$

κ = 0, central difference extrapolation between adjacent volumes.

$$u^L_{i+½} = u_i + \tfrac{1}{2}(u_{i+1} - u_{i-1}) \tag{2.28}$$

$$u^R_{i-½} = u_{i+1} - \tfrac{1}{2}(u_{i+2} - u_i) \tag{2.29}$$

Slope limiters

Replacing constant piecewise data reconstruction with linear piecewise reconstruction results in overshoots of the numerical solution about discontinuities in the flow field (Godunov's theorem, Toro [1]). To avoid overshoots, limiters are applied to restrict the extrapolated values so that they remain in the range between the left and right volumes. In regions of smooth flow limiters allow higher order spatial accuracy but, in regions of sharp gradients they locally reduce the accuracy towards first order, thus avoiding overshoots in the solution. If a suitable limiter Φ(r), to be introduced later in this section, is added to the extrapolation then Equations (2.24) and (2.25) become.

$$u^L_{i+½} = u_i + \tfrac{1}{4}\Phi(r^+_{i-½})(1-\kappa)(u_i - u_{i-1}) + \tfrac{1}{4}\Phi(r^-_{i+½})(1+\kappa)(u_{i+1} - u_i) \tag{2.30}$$

$$u^{R}_{i-\frac{1}{2}} = u_i - \tfrac{1}{4}\Phi(r^{-}_{i+\frac{3}{2}})(1+\kappa)(u_i - u_{i-1}) - \tfrac{1}{4}(1-\kappa)\Phi(r^{+}_{i+\frac{1}{2}})(u_{i+1} - u_i) \tag{2.31}$$

The superscripts, '+' and '-', refer to positive and negative waves respectively. At the i-$^1/_2$ face waves approaching from the left are considered whilst at the i+$^1/_2$ face waves travelling from the right are considered.

$$r^{+}_{i-\frac{1}{2}} = \frac{u_{i+1} - u_i}{u_i - u_{i-1}} \tag{2.32}$$

$$r^{-}_{i+\frac{1}{2}} = \frac{u_i - u_{i-1}}{u_{i+1} - u_i} \tag{2.33}$$

if we let

$$r^{L} = \frac{u_{i+1} - u_i}{u_i - u_{i-1}} \tag{2.34}$$

then equation (2.30) can be written as

$$u^{L}_{i+\frac{1}{2}} = u_i + \tfrac{1}{2}\Psi^{L}(u_i - u_{i-1}) \tag{2.35}$$

where

$$\Psi = \tfrac{1}{2}\left[(1-\kappa)\Phi(r^{L}) + (1+\kappa)r^{L}\Phi(\frac{1}{r^{L}})\right] \tag{2.36}$$

Following a similar argument the right side data state is given by.

$$u^{R}_{i+\frac{1}{2}} = u_i + \tfrac{1}{2}\Psi^{r}(u_{i+2} - u_{i+1}) \tag{2.37}$$

where

$$\Psi = \tfrac{1}{2}\left[(1-\kappa)\Phi(r^{R}) + (1+\kappa)r^{R}\Phi(\frac{1}{r^{R}})\right] \tag{2.38}$$

$$r^{R} = \frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} \tag{2.39}$$

Van Leer slope limiter

Now the form of the slope limiter function, Φ(r), used in the previous section is defined. There are various forms of limiter functions available some of which have been outlined in the texts by Hirsch [2] and Toro [1]. The limiter selected was a Van Leer limiter which is illustrated in Figure 2-6. The Van Leer limiter function is defined:

$$\Phi(r) = \begin{cases} 0, & r \le 0 \\ \dfrac{2r}{1+r}, & r > 0 \end{cases} \tag{2.40}$$

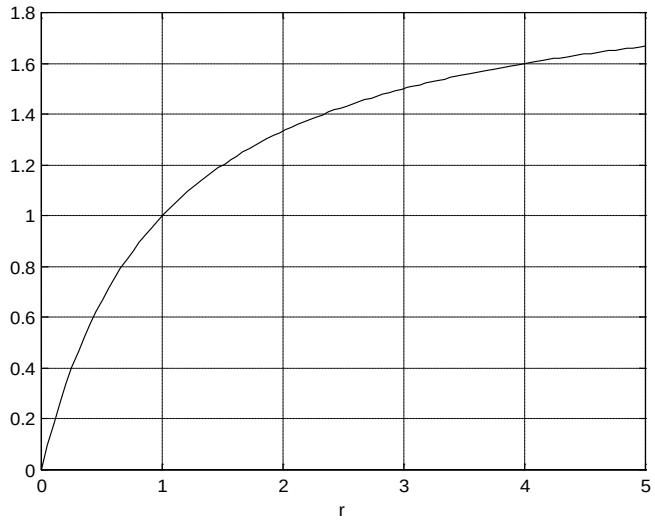


Figure 2-6. Van Leer slope limiter function.

Heat transfer and pipe friction

As the gas dynamics code is explicitly integrated in time, the inclusion of heat transfer and pipe friction require the estimation of the instantaneous heat transfer and pipe friction coefficients at each point in time of the integration scheme.

Heat loss to the pipe wall was calculated using the empirical Dittus-Boelter equation assuming fully developed turbulent flow. The Dittus-Boelter equation yields the Nusselt number from which the heat transfer coefficient can be approximated.

$$Nu = 0.0023\,\mathrm{Re}^{0.8}\,\mathrm{Pr}^{0.3} \tag{2.41}$$

Nu = Nusselt number

Re = Reynolds's number

Pr = Prandlt number

The mean heat transfer coefficient is then given by

$$h = \frac{Nu \times k}{d} \tag{2.42}$$

h = heat transfer coefficient

k = thermal conductivity

d = mean pipe diameter

For these calculations it was necessary to have appropriate values for the thermal conductivity, Prandlt number and viscosity, each of which vary with temperature and composition of the gas. These were approximated with the use of the polynomials given in Blair [3].

Pr = 0.71

$$k = 0.0061944 + 7.3814 \times 10^{-5} T - 1.2491 \times T^2 \quad \text{(W/mK)} \tag{2.43}$$

$$\mu = 7.457 + 4.1547 \times 10^{-6} T - 7.4793 \times 10^{-8} T^2 \quad \text{(kg/ms)} \tag{2.44}$$

The friction factor at the pipe wall was approximated by Blasius's empirical formula assuming fully developed turbulent flow in a smooth pipe. The friction factor, '*f*' is given by:

$$f = 0.0791\,\mathrm{Re}^{-\frac{1}{4}} \tag{2.45}$$

where Re is the Reynolds number defined:

$$\mathrm{Re} = \frac{\rho u d}{\mu} \tag{2.46}$$

where d is the pipe diameter and μ is viscosity

The net friction can then be calculated from the average shear stress at the wall, τ, which is given by:

$$\tau = f \tfrac{1}{2} \rho u^2 \tag{2.47}$$

The assumption of fully developed turbulent flow in both the heat transfer and pipe friction calculations is not particularly accurate, however it was necessary to establish simple predictive empirical relations for the respective instantaneous coefficients. In reality the pipe flow is highly unsteady within the time scale of the time-stepping scheme and assuming fully developed turbulent flow was a simplification. The alternative would be to account for the complex unsteady flows in the gas dynamics of engine manifolds or adopt some other approximate method, both of which have been left for future research.

Species tracking

The composition of the gas in pipe flow is made up from a mixture of 12 chemical species and gaseous fuel. Fresh air, air/fuel mixture and burnt products are represented by the appropriate combination of these 12 species and fuel. To model the propagation of species in the pipe a species conservation equation, in addition to the conservation of mass, energy and momentum, has been implemented. The mass, energy and momentum equation are first updated at a new time step then the auxiliary species conservation equation is solved using the updated mean density from the mass equation.

If $x(s)$ is the mass fraction of species, s, then the update of the species mass fraction, at the time, $j+1$, is of the form:

$$\rho_i^{j+1} x_i^{j+1}(s) = \rho_i^j x_i^j(s) - \frac{\Delta t}{V}\left[\left(x(s) \times \rho u A\right)^j_{i+\frac{1}{2}} - \left(x(s) \times \rho u A\right)^j_{i-\frac{1}{2}} \right] \tag{2.48}$$

where ρ represents the mean density for all species.

The mean density at the new time step, ρ_i^{j+1}, is already known from the mass conservation equation, therefore:

$$x_i^{j+1}(s) = \frac{\rho_i^j x_i^j(s) - \frac{\Delta t}{V}\left[\left(x(s)\times \rho u A\right)_{i+\frac{1}{2}}^j - \left(x(s)\times \rho u A\right)_{i-\frac{1}{2}}^j\right]}{\rho_i^{j+1}} \tag{2.49}$$

Once the species fraction update was performed the species mass fractions were normalised to ensure the mass fractions always summed to one. The mass fractions at each face were taken as the average of those in the adjacent volumes. The form of the species transport considers convection only and does not include diffusion terms. In most IC engines the pipe velocity is sufficiently high that convection is dominant and it is safe to neglect diffusion. Equation (2.49) was then integrated in time using a modified Runge-Kutta technique similar to the method given in the time stepping section of this chapter.

The introduction of variable composition required a modification to the Riemann solution so that the left and right side wave calculations use different values for the ratio of specific heats, due to variation of composition in the left and right initial conditions. The specific heat capacity of each species in the flow has been assumed constant, having been evaluated at a nominated temperature before the simulation starts. A natural extension for future work would be to include the variation of heat capacity with temperature into the pipe simulation.

Fuel injection

In addition to the conservative fluxes at either end of each pipe volume, any pipe volume may have addition fluxes that represent a fuel injector in the pipe. The fuel injector is represented by an addition fuel mass flux and energy flux to the volume. The fuel mass flux is calculated so that the air/fuel ratio in the volume is maintained at the predetermined level set in the input file. It is assumed that the fuel instantly vaporises as it enters the pipe and the fuel energy flux is set to equal the latent heat of vaporisation of the fuel. It has been assumed that the fuel's momentum component along the pipe is nominally zero and can be ignored.

Time stepping

An explicit four stage modified Runge-Kutta technique was used to integrate in time. A time step update was made using the following algorithm:

$$\overline{W}_i^1 = W_i$$

$$\overline{W}_i^{j+1} = W_i - \alpha_j \frac{\Delta t}{V}\left[F(\overline{W}^j)_{i+\frac{1}{2}} - F(\overline{W}^j)_{i-\frac{1}{2}} + G(\overline{W}^j)_i\right]$$

$$j = 1,2,3,4 \tag{2.50}$$

$$\alpha = [¼, ⅓, ½, 1]$$

$$W_{i+1} = \overline{W}_i^5$$

The subscript, *i*, refers to the main time step index and the subscript, *j*, refers to the sub-step index.

The magnitude of the time step is given by

$$\Delta t = \frac{CFL \times \delta x}{S} \tag{2.51}$$

Where, *S*, is the maximum magnitude of the propagation speed in the volume, δx, is the minimum volume length and CFL is the Courant-Freidreich-Lewys number. The exact Riemann solution yields the speed of the left and right travelling waves on each side of a volume face. The maximum magnitude of these gives the value, *S*. For simple pipe simulations the maximum theoretical CFL of $2\sqrt{2}$ could be used however for more complicated engine simulations a rather conservative value of 0.6 was required to ensure stability.

Discussion

The numerical scheme has produced good accuracy and resolution of discontinuities in the flow field. Figure 2-7 shows the steady state solution for flow through a nozzle with a shock compared to the analytic solution for the nozzle. The nozzle specification is the same as that used by Anderson [4]. The cross-sectional area as a function of the distance through the nozzle is given by:

$$A(x) = 2.2x^2 - 6.6x + 5.95 \qquad 0 < x < 3 \tag{2.52}$$

The upstream stagnation pressure and temperature were 101325 Pa and 300K respectively, with the nozzle back pressure set to give a pressure ratio across the nozzle of 0.7. The numerical solution is free from oscillations about the shock and accurately locates the shock position, validating the numerical simulation for this type of flow.

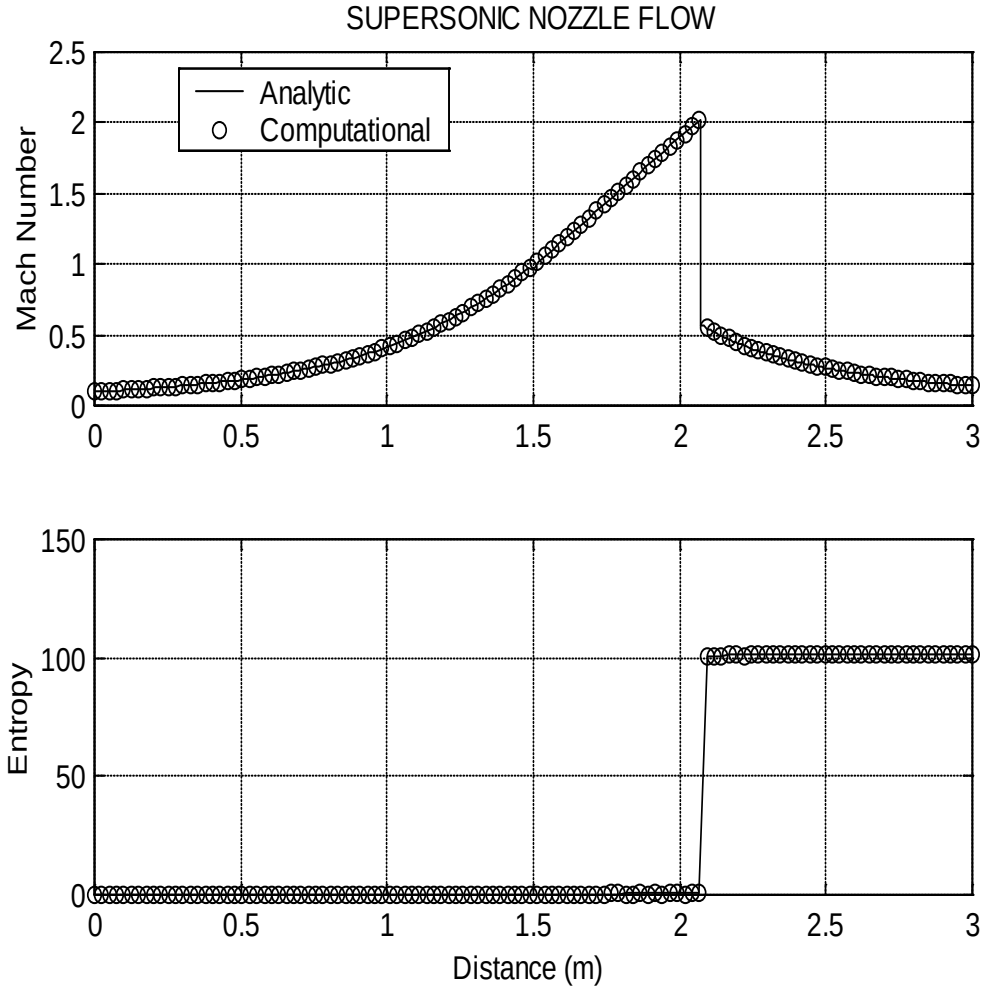


Figure 2-7. Nozzle flow.

Simple empirical relations have been used to approximate the heat transfer and friction in the pipe. The numerical scheme accurately models these effects and gives excellent agreement with the analytical cases for Rayleigh and Fanno flows as shown in Figures 2-8 and 2-9. Provided the heat transfer coefficient and pipe friction factor are correctly estimated the numerical code accurately models the physical effects of both friction and heat transfer on pipe flow.

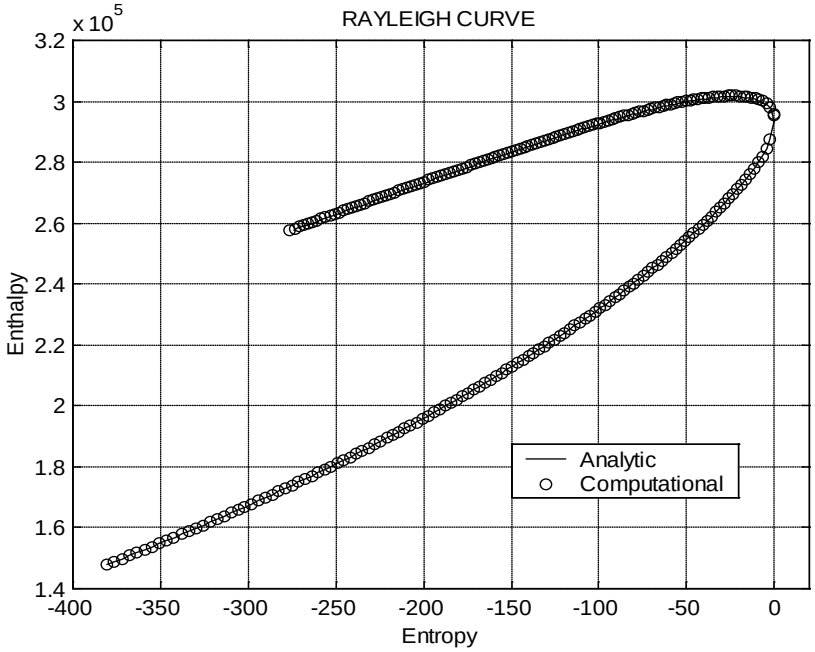


Figure 2-8. Rayleigh curve for heat transfer.

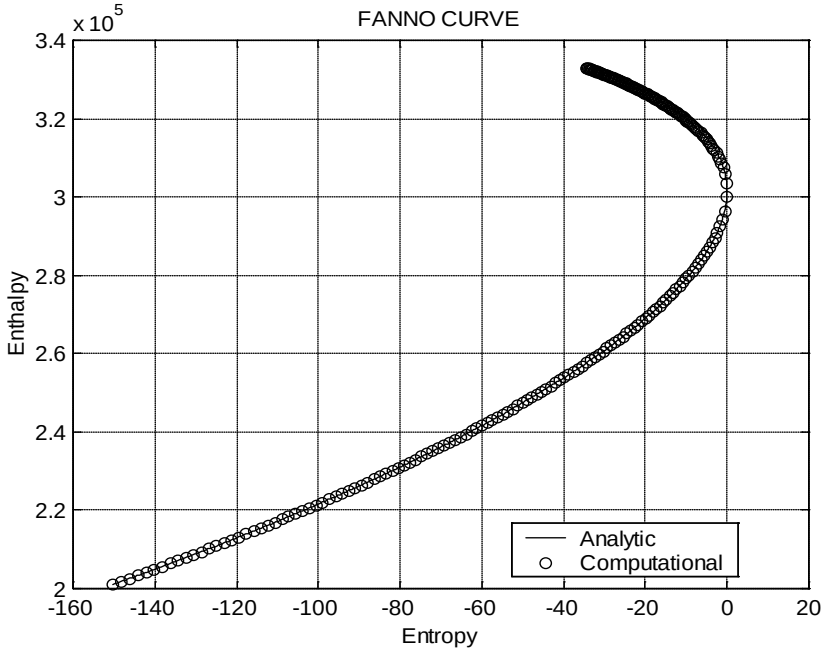


Figure 2-9. Fanno curve for pipe friction.

The most significant factor for accurate modelling is how well the empirical relations estimate the pipe friction and heat transfer. The Blasius's and Dittus-Boelter relations, for pipe friction and heat transfer respectively, have been developed for steady flow in pipes however it has been assumed they are a reasonable first approximation for unsteady flow. This approach gives reasonable estimation of friction and heat transfer and is adequate for the majority of engine simulations. For situations where heat transfer and friction have a more important effect on engine performance e.g. turbocharged engines, then more rigorous models may be required. These models could also be further improved by calculating the thermal conductivity and viscosity of the fluid as functions of composition as well as temperature.

Figure 2-10 shows the analytic and numerical results for the shock tube problem with discontinuous composition in the left and right states. The numerical and analytic solution correlate closely indicating that the numerical scheme has correctly captured the physics of the Riemann problem and produces time accurate results. The problem was set up so that the left side composition represents completely burnt air-fuel mixture at high pressure and the right side represents air at low pressure. The input file for this shock tube problem is given in the appendix. For pure convection the discontinuity in composition should remain in the solution and convect with the contact discontinuity. The plot of concentration indicates that the linear interpolation of composition to the volume face introduces numerical diffusion into the scheme, resulting in the concentration discontinuity being smeared out. Whilst diffusion modelling has not intentionally been added a small amount is physically realistic.

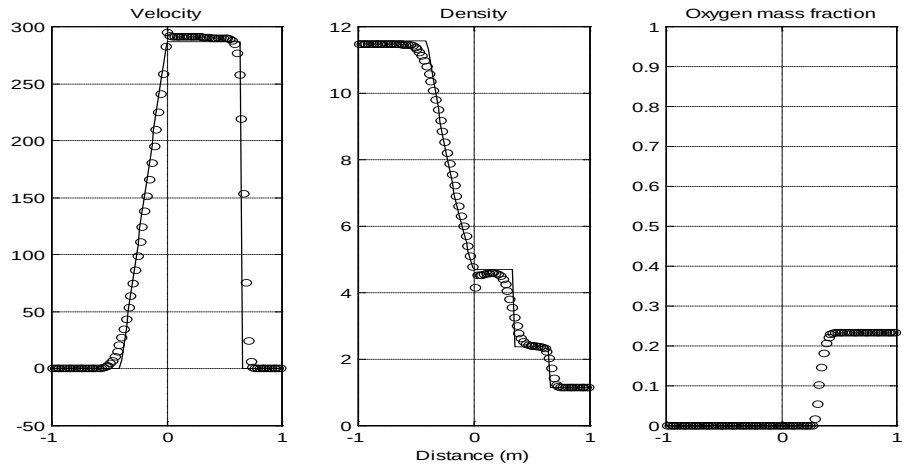


Figure 2-10. Numerical solution to the shock tube problem with discontinuous gas properties (100 volumes).

conclusion

Overall the numerical gas dynamics code correlates closely with the standard analytic test cases and produces accurate results. The code resolves discontinuities in the flow field without overshoots whilst maintaining 2^{nd} order accuracy in regions of smooth flow. Time accuracy has also been proven by correlation with the shock tube problem. The effect of heat transfer and pipe friction on the pipe flow are physically correct but future work may seek to better approximate the heat transfer coefficient and pipe friction factor. Propagation of species in the pipe has been verified to a limited extent by correlation with the shock tube problem. A recommendation for future development is to model the variation of the specific heat capacity of the gas with temperature in exhaust pipe flows where relatively large temperature gradients exist.

References

[1] Toro, E. F. (1997) *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer, New York.

[2] Hirsch. C. (1990) *Numerical computation of internal and external flows*, John Wiley & Sons, England, Vol 2

[3] Blair, G. P. (1996) *Design and simulation of two-stroke engines*, SAE, Warrendale.

[4] Anderson, J. D. (1995) *Computational fluid dynamics:the basics with applications*, McGraw-Hill, New York

3. VALVE FLOW

The gas dynamics used to model pipe elements in the engine simulation rely on the correct application of boundary conditions. At each boundary these may be either, an open or closed end, or a valve with a throat. Added to this is the possibility of inflow and outflow, in either case being either subsonic or choked sonic flow. All of these boundary conditions can occur within a single pipe element as a simulation progresses, and need to be included in the boundary condition model. The text by Benson [1], and the more recent update by Winterbone and Pearson [2], outline standard methods for boundary condition treatment in engine simulation.

In gas dynamics information propagates through the pipe along characteristics (waves). Any attempt to correctly model boundary conditions must include this characteristic information as it enters and leaves the flow domain at the boundary. Hirsch [4] shows that the imposed boundary conditions need to satisfy the compatibility equations or Riemann invariants for the outgoing characteristics. Well posed boundary conditions will satisfy the propagation of characteristic information in the pipe.

Riemann Variables

The implementation of boundary conditions closely follows the methods introduced by Vandevoorde Et al [4]. Figure 3-1 schematically shows the paths along which characteristic information, or Riemann invariants, propagate in gas dynamics, where u is the velocity and c is the local speed of sound.

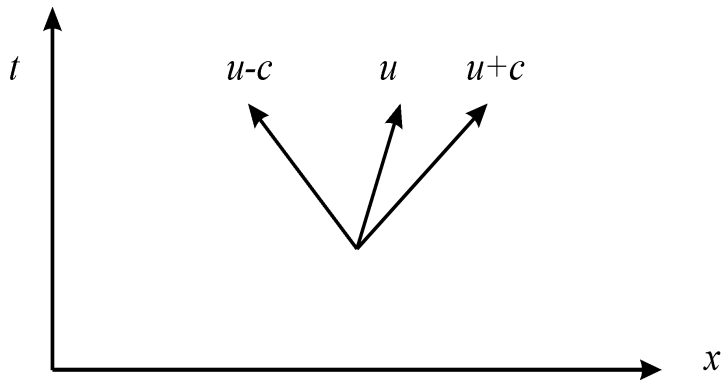


Figure 3-1. Schematic of Riemann invariant path lines in one dimension.

The Riemann invariants along the characteristic paths *u+c*, *u*, and *u-c* are given by:

Riemann invariant along *u-c* characteristic

$$\delta u = \rho c \delta p \tag{3.1}$$

where p is the pressure and ρ is the density

Riemann invariant along *u* characteristic, (isentropic)

$$s = const$$

where *s* is entropy (3.2)

Riemann invariant along *u+c* characteristic

$$\delta u = -\rho c \delta p \tag{3.3}$$

Integrating Equations, (3.1) and (3.3)

between the old state at the pipe boundary and the new updated boundary condition gives:

u-c characteristic

$$u_{new} - u_{old} = \frac{2c_{old}}{\gamma - 1}\left[\frac{p_{new}}{p_{old}}^{\frac{(\gamma-1)}{2\gamma}} - 1\right] \tag{3.4}$$

u+c characteristic

$$u_{new} - u_{old} = -\frac{2c_{old}}{\gamma - 1}\left[\frac{p_{new}}{p_{old}}^{\frac{(\gamma-1)}{2\gamma}} - 1\right] \tag{3.5}$$

Both the differential form and integrated form of the Riemann invariants are useful, where appropriate, to calculate boundary conditions.

Along the *u* characteristic isentropic conditions give:

$$\frac{\rho_{new}}{\rho_{old}} = \left(\frac{p_{new}}{p_{old}}\right)^{\frac{1}{\gamma}} \tag{3.6}$$

Inflow

Subsonic pipe inflow with no throat.

In this section the boundary conditions for subsonic flow into a pipe, and the extension for when sonic flow occurs at the boundary, are introduced. The subscript ‘res’, ‘pipe’ and ‘n’ indicate state variables in the reservoir, pipe and boundary condition respectively. Figure 3-2 shows a schematic diagram of flow into the open end of a pipe.

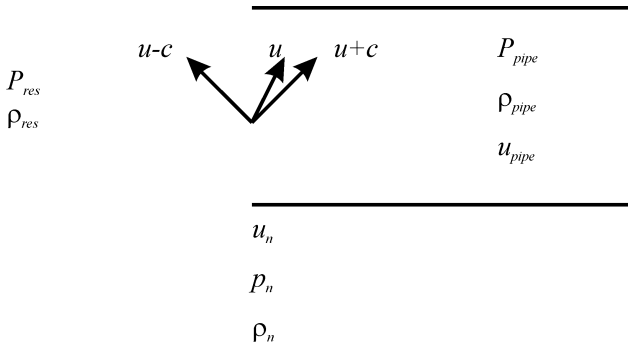


Figure 3-2. Schematic of flow into the open end of a Pipe.

Applying conservation of energy,

$$u_n^{\,2} = \frac{2}{\gamma - 1} c_{res}^2 \left[1 - \frac{p_{new}}{p_{res}}^{\frac{(\gamma-1)}{\gamma}} \right] \tag{3.7}$$

Integrated Riemann invariant along u-c, (Equation (3.4)).

$$u_n - u_{pipe} = \frac{2c_{pipe}}{\gamma - 1} \left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}} - 1 \right] \tag{3.8}$$

Isentropic flow from reservoir to pipe.

$$\frac{\rho_n}{\rho_{res}} = \left(\frac{p_n}{p_{res}} \right)^{\frac{1}{\gamma}} \tag{3.9}$$

Squaring the Riemann invariant and collecting terms gives:

$$u_n{}^2 = \left(u_{pipe} - \frac{2}{(\gamma-1)}c_{pipe}\right)^2 + \left(u_{pipe} - \frac{2}{(\gamma-1)}c_{pipe}\right)\frac{1}{(\gamma-1)}c_{pipe}\left(\frac{P_n}{p_{pipe}}\right)^{\frac{(\gamma-1)}{2\gamma}} + \left(\frac{2}{(\gamma-1)}\right)^2 c_{pipe}^2\left(\frac{P_n}{p_{pipe}}\right)^{\frac{(\gamma-1)}{\gamma}} \tag{3.10}$$

Equating Equation (3.10) to Equation (3.7) and collecting terms leads to a quadratic equation for the subsonic flow into a pipe through a valve throat:

$$Ax^2 + Bx + C = 0 \tag{3.11}$$

where

$$A = \left(\frac{2}{(\gamma-1)}\right)^2 c_{pipe}^2\left(\frac{1}{p_{pipe}}\right)^{\frac{(\gamma-1)}{\gamma}} + \frac{2}{(\gamma-1)}c_{res}^2\left(\frac{1}{p_{res}}\right)^{\frac{(\gamma-1)}{\gamma}}$$

$$B = \left(u_{pipe} - \frac{2}{(\gamma-1)}c_{pipe}\right)\frac{1}{(\gamma-1)}c_{pipe}\left(\frac{1}{p_{pipe}}\right)^{\frac{(\gamma-1)}{2\gamma}}$$

$$C = \left(u_{pipe} - \frac{2}{(\gamma-1)}c_{pipe}\right)^2 - \frac{2}{(\gamma-1)}c_{res}^2$$

$$x = p_n{}^{\frac{(\gamma-1)}{2\gamma}} \tag{3.12}$$

At the sonic limit the pipe will choke at the boundary and the boundary condition will be dependant on the state of the reservoir only, leading to:

$$p_n = p_{res}\left(\sqrt{\frac{\gamma+1}{2}}\right)^{\frac{2\gamma}{\gamma-1}}$$

$$\frac{\rho_n}{\rho_{res}} = \left(\sqrt{\frac{\gamma+1}{2}}\right)^{\frac{2}{\gamma-1}}$$

$$u_n = \sqrt{\gamma\frac{P_n}{u_n}} \tag{3.13}$$

Pipe inflow through a valve throat

For flow into a pipe, through a throat, it is assumed that there is no pressure recovery after the valve throat so that pressure at the throat is the same as the applied boundary condition, i.e. isobaric. This assumption has been made because the flow downstream of a valve throat is highly turbulent and irreversible. The large losses result in a small amount of pressure recovery. The assumption of isobaric flow simplifies the calculations and gives a reasonable approximation of the flow. In reality a small amount of pressure recovery does occur so that the valve model moderately underestimates the flow though a valve. The subscript *t* refers to flow variables at the effective valve throat and *A* is cross-section area.

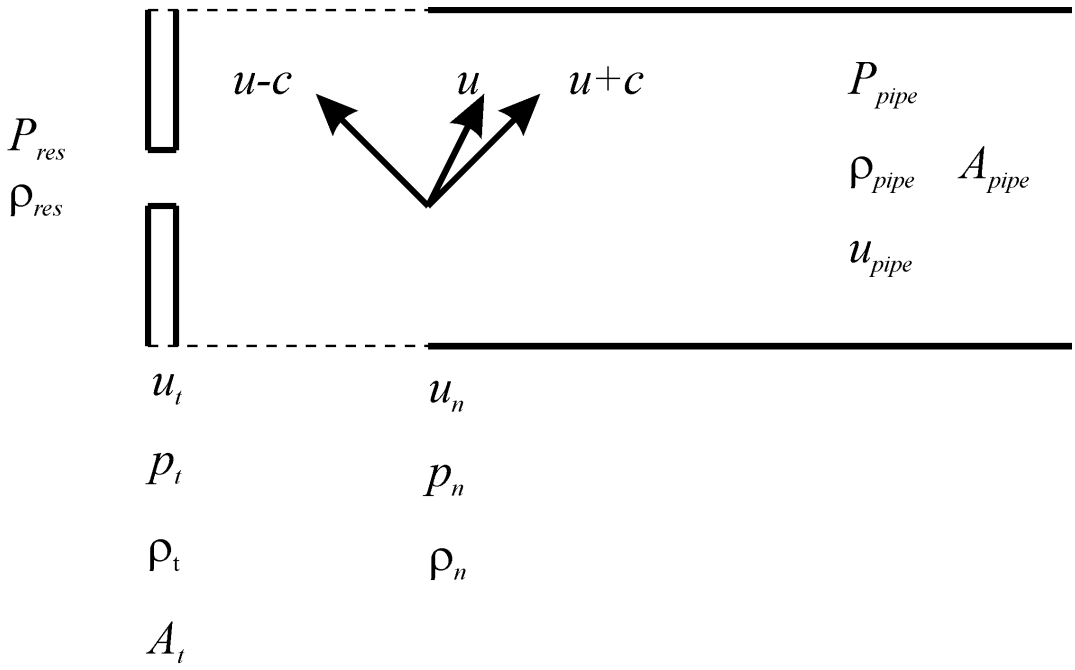


Figure 3-3. Schematic of subsonic inflow to a pipe through a valve throat.

Applying conservation of energy

$$\frac{\gamma}{(\gamma-1)}\frac{p_{res}}{\rho_{res}} = \frac{\gamma}{(\gamma-1)}\frac{p_t}{\rho_t} + \frac{1}{2}u_t^{\,2} \tag{3.14}$$

$$a_{res}^{\,2} = a_n^{\,2} + \frac{\gamma-1}{\gamma}u_n^{\,2} \tag{3.15}$$

Conservation of mass

$$\rho_n u_n A_n = \rho_t u_t A_t \tag{3.16}$$

Isentropic from reservoir to throat

$$\frac{\rho_t}{\rho_{res}} = \left(\frac{p_t}{p_{res}} \right)^{\frac{1}{\gamma}} \tag{3.17}$$

Isobaric from throat to pipe

$$p_t = p_n \tag{3.18}$$

Riemann invariant

$$u_n - u_{pipe} = \frac{2c_{pipe}}{\gamma - 1} \left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}} - 1 \right] \tag{3.19}$$

from Equation (3.17) and Equation (3.18)

$$\frac{1}{\rho_t} = \frac{1}{\rho_{res}} \left(\frac{p_{res}}{p_n} \right)^{\frac{1}{\gamma}} \tag{3.20}$$

from Equation (3.20) and Equation (3.14)

$$\frac{c_{res}^{\,2}}{(\gamma - 1)} \left(1 - \left(\frac{P_n}{p_{res}} \right)^{\frac{\gamma-1}{\gamma}} \right) = \frac{\gamma}{(\gamma - 1)} \frac{p_t}{\rho_t} + \frac{1}{2} u_t^{\,2} \tag{3.21}$$

now using conservation of mass, Equation (3.16)

$$\frac{c_{res}^{\,2}}{(\gamma - 1)} \left(1 - \left(\frac{P_n}{p_{res}} \right)^{\frac{\gamma-1}{\gamma}} \right) = \tfrac{1}{2} u_n^{\,2} \frac{\rho_n^{\,2} A_n^{\,2}}{\rho_t^{\,2} A_t^{\,2}} \tag{3.22}$$

we know

$$\frac{\rho_n}{\rho_t} = \frac{p_n}{p_t} \left(\frac{c_t}{c_n} \right)^2 = \left(\frac{c_t}{c_n} \right)^2 \tag{3.23}$$

and

$$\frac{c_t}{c_n} = \frac{c_t}{c_{res}} \frac{c_{res}}{c_n} \tag{3.24}$$

also

$$\frac{a_t}{a_{res}} = \left(\frac{p_t}{p_{res}}\right)^{\frac{\gamma-1}{2\gamma}} = \left(\frac{p_n}{p_{res}}\right)^{\frac{\gamma-1}{2\gamma}} \tag{3.25}$$

and from Equation (3.15)

$$\frac{{c_n}^2}{{c_{res}}^2} = 1 - \frac{\gamma-1}{2}\frac{{u_n}^2}{{c_{res}}^2} \tag{3.26}$$

combining Equation (3.22) and Equation (3.26)

$$\frac{{c_{res}}^2}{(\gamma-1)}\left(\left(\frac{P_{res}}{p_n}\right)^{\frac{\gamma-1}{\gamma}} - 1\right) = \frac{\frac{1}{2}{u_n}^2\frac{{A_n}^2}{{A_t}^2}}{\left(1 - \frac{\gamma-1}{2}\frac{{u_n}^2}{{c_{res}}^2}\right)^2} \tag{3.27}$$

Equation (3.27) along with the Riemann invariant, Equation (3.19), gives two equations in two unknowns that can be solved for u_n and p_n using a Newton-Rhapson technique. The final variable ρ_n can then be calculated from mass continuity.

Equation (3.27) can be recast in the form.

$$\begin{aligned}
F &= \frac{{c_{res}}^2}{(\gamma-1)}A - \tfrac{1}{2}\frac{{A_n}^2}{{A_t}^2}B \\
A &= \left(\left(\frac{P_{res}}{p_n}\right)^{\frac{\gamma-1}{\gamma}} - 1\right) \\
B &= \frac{C}{D} \\
C &= {u_n}^2 \\
u_n &= u_{pipe} + \frac{2c_{pipe}}{\gamma-1}\left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}} - 1\right] \\
D &= \left(1 - \frac{\gamma-1}{2}\frac{{u_n}^2}{{c_{res}}^2}\right)^2
\end{aligned} \tag{3.28}$$

The system equations in terms of the Newton-Raphson iterative technique than take the form.

$$p_n^{\,i+1} = p_n^{\,i} - \frac{F}{\frac{dF}{dP_n}} \tag{3.29}$$

$$\frac{dF}{dP_n} = \frac{\partial F}{\partial A}\frac{\partial A}{\partial p_n} + \frac{\partial F}{\partial B}\left(\frac{\partial B}{\partial C}\frac{\partial C}{\partial u_n}\frac{\partial u_n}{\partial p_n} + \frac{\partial B}{\partial D}\frac{\partial D}{\partial u_n}\frac{\partial u_n}{\partial p_n}\right) \tag{3.30}$$

Pipe inflow through a choked throat.

Once the throat becomes choked the flow properties at the throat are dependant on the state of the reservoir only. The boundary conditions however remain subsonic until the pressure in the pipe falls sufficiently low for fully supersonic flow to occur. For choked flow it is therefore necessary to include the characteristic information travelling out of the pipe within the boundary condition calculations. The transition from choked flow to fully supersonic flow involves the input of a non-stationary shock wave as a boundary condition. In practice it is difficult to calculate the strength of this shock, and as it occurs rarely in normal operation of IC engines, has been omitted from the valve model.

Conservation of energy

$$\frac{\gamma}{(\gamma-1)}\frac{p_{res}}{\rho_{res}} = \frac{\gamma}{(\gamma-1)}\frac{p_n}{\rho_n} + \frac{1}{2}u_n^{\,2} \tag{3.31}$$

Conservation of mass

$$\rho_{pipe}u_{pipe}A_{pipe} = \rho_n u_n A_n \tag{3.32}$$

Differential form of the Riemann invariant along (*u*-*c*)

$$\delta p - \rho c\,\delta u = 0 \tag{3.33}$$

In discrete form assuming a small variation in state variables Equation (3.33) is approximated by,

$$(p_n - p_{pipe}) - \rho_{pipe}c_{pipe}(u_n - u_{pipe}) = 0 \tag{3.34}$$

from Equation (3.31) and Equation (3.32)

$$\frac{\gamma}{(\gamma-1)}\frac{p_{res}}{\rho_{res}} = \frac{\gamma}{(\gamma-1)}\frac{A_{pipe}}{\rho_t u_t A_t}u_n P_n + \frac{1}{2}u_n^{\,2} \tag{3.35}$$

from Equation (3.33) and Equation (3.35) after rearrangement leads to the quadratic

$$\left(\frac{A_{pipe}\rho_{pipe}c_{pipe}}{\rho_t u_t A_t}+\frac{\gamma-1}{2\gamma}\right)u_n{}^2+\frac{A_{pipe}}{\rho_t u_t A_t}\left(p_{pipe}-\rho_{pipe}c_{pipe}u_{pipe}\right)u_n-\frac{P_{res}}{\rho_{res}}=0 \tag{3.36}$$

Where the variables at the choked throat are given by.

$$p_t = p_{res}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{3.37}$$

$$\frac{\rho_t}{\rho_{res}}=\left(\sqrt{\frac{\gamma+1}{2}}\right)^{\frac{2}{\gamma-1}} \tag{3.38}$$

$$u_t=\sqrt{\gamma\frac{P_t}{u_t}} \tag{3.39}$$

Outflow

Pipe outflow with no throat

Out flow from a pipe open end is calculated using the Riemann invariants leaving the pipe and the assumption that pressure at the end of the pipe equals that in the reservoir. Figure 3-4 shows a schematic diagram of flow out of the open end of a pipe.

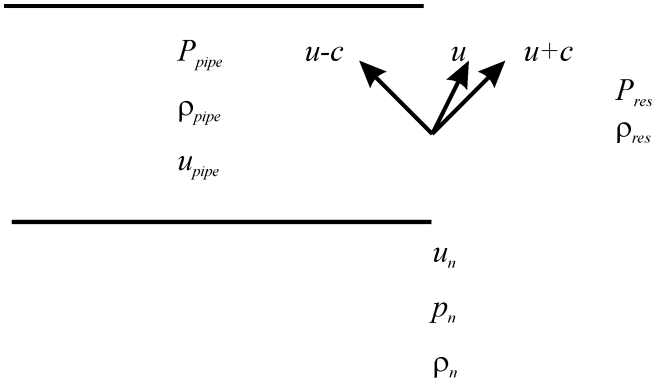


Figure 3-4. Schematic of outflow from a pipe without a throat.

Riemann invariant

$$u_n-u_{pipe}=-\frac{2c_{pipe}}{\gamma-1}\left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}}-1\right] \tag{3.40}$$

The flow out of the pipe can be assumed to be a free jet therefore the boundary condition pressure can be taken as the pressure in the reservoir.

$$p_n = p_{res} \tag{3.41}$$

Isentropic from the pipe to the boundary condition.

$$\frac{\rho_n}{\rho_{pipe}} = \left(\frac{p_n}{p_{pipe}} \right)^{\frac{1}{\gamma}} \tag{3.42}$$

Equations (3.40) to (3.42) fully describe the boundary condition calculations. When the sonic point is reached all three of the characteristics leave the pipe boundary and the boundary condition becomes independent of the state of the reservoir. The boundary conditions are then given by:

$$\begin{aligned} p_n &= p_{pipe} \\ \rho_n &= \rho_{pipe} \\ u_n &= u_{pipe} \end{aligned} \tag{3.43}$$

Subsonic pipe outflow through valve throat

For flow out of a pipe, through a throat, it is assumed that the flow in the reservoir is a free jet, therefore the pressure at the effective valve throat is equal to the pressure in the reservoir.

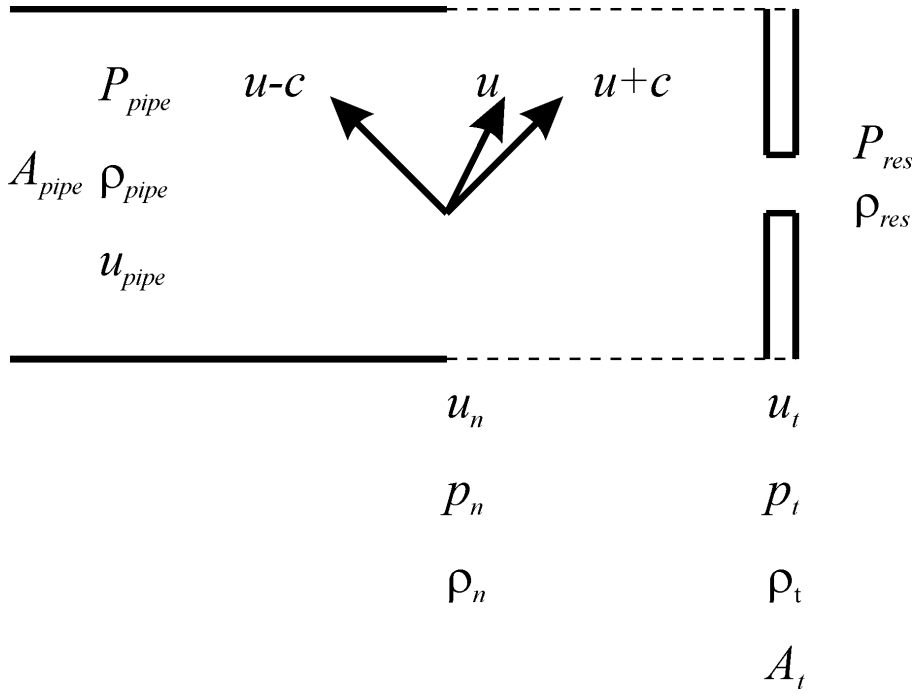


Figure 3-5. Schematic of outflow from a pipe through a valve throat.

Conservation of energy

$$\frac{\gamma}{(\gamma-1)}\frac{p_n}{\rho_n}+\frac{1}{2}u_n=\frac{\gamma}{(\gamma-1)}\frac{p_t}{\rho_t}+\frac{1}{2}u_t^{\,2} \tag{3.44}$$

Conservation of mass

$$\rho_t u_t A_t=\rho_n u_n A_n \tag{3.45}$$

Isentropic flow from pipe to throat

$$\frac{\rho_p}{\rho_t}=\left(\frac{p_p}{p_t}\right)^{\frac{1}{\gamma}}$$
$$\frac{\rho_p}{\rho_n}=\left(\frac{p_p}{p_n}\right)^{\frac{1}{\gamma}} \tag{3.46}$$

Free jet in reservoir

$$p_t=p_{res} \tag{3.47}$$

Riemann invariant

$$u_n-u_{pipe}=-\frac{2c_{pipe}}{\gamma-1}\left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}}-1\right] \tag{3.48}$$

using conservation of mass, Equation (3.45), and the isentropic law, Equation (3.46).

$$u_t=\frac{u_n\rho_n A_n}{\rho_t A_t}=\frac{u_n A_n}{A_t}\left(\frac{p_n}{p_t}\right)^{\frac{1}{\gamma}} \tag{3.49}$$

now from Equations (3.44), (3.46) and (3.49).

$$\frac{\gamma}{(\gamma-1)}\frac{p_{pipe}^{\frac{1}{\gamma}}}{\rho_{pipe}}p_n^{\frac{\gamma-1}{\gamma}}+\tfrac{1}{2}u_{pipe}=\frac{\gamma}{(\gamma-1)}\frac{p_{pipe}^{\frac{1}{\gamma}}}{\rho_{pipe}}p_t^{\frac{\gamma-1}{\gamma}}+\tfrac{1}{2}u_n^{\,2}\frac{A_n^{\,2}}{A_t^{\,2}}\left(\frac{p_n}{p_t}\right)^{\frac{2}{\gamma}} \tag{3.50}$$

rearranging gives

$$\frac{\gamma}{(\gamma-1)}\frac{p_{pipe}^{\frac{1}{\gamma}}}{\rho_{pipe}}\left(p_n^{\frac{\gamma-1}{\gamma}}-p_t^{\frac{\gamma-1}{\gamma}}\right)+\tfrac{1}{2}u_{pipe}\left(1-\phi\left(\frac{p_n}{p_t}\right)^{\frac{2}{\gamma}}\right)=0 \tag{3.51}$$

where

$$\phi=\left(\frac{A_n}{A_t}\right)^2$$

Equation (3.51) can be written in the form

$$F = A\left(p_n^{\frac{\gamma-1}{\gamma}} - p_t^{\frac{\gamma-1}{\gamma}} \right) + \tfrac{1}{2} B u_n^{\,2} \tag{3.52}$$

Where

$$A = \frac{\gamma}{\gamma - 1} \frac{p_{pipe}^{\frac{1}{\gamma}}}{\rho_{pipe}}$$

$$B = 1 - \phi \left(\frac{p_n}{p_t} \right)^{\frac{2}{\gamma}} \tag{3.53}$$

The above form of the system equation is amenable to solution by the Newton-Raphson technique, of the form.

$$\frac{dF}{dp_n} = \frac{\gamma - 1}{\gamma} A p_n^{-\frac{1}{\gamma}} + B u_n \frac{\partial u_n}{\partial p_n} + \tfrac{1}{2} u_n^{\,2} \frac{\partial B}{\partial P_n} \tag{3.54}$$

$$p_n^{\,i+1} = p_n^{\,i} - \frac{F}{\frac{dF}{dP_n}}$$

Pipe outflow through a choked throat

Once the sonic point has been reached at the throat the flow becomes independent of the pressure in the reservoir. The boundary conditions are calculated from the state of the pipe only. The case for supersonic flow upstream of the valve throat has been omitted.

Conservation of energy

$$\frac{\gamma}{(\gamma - 1)} \frac{p_{pipe}}{\rho_{pipe}} + \frac{1}{2} u_{pipe}^2 = \frac{\gamma}{(\gamma - 1)} \frac{p_n}{\rho_n} + \frac{1}{2} u_n^{\,2} \tag{3.55}$$

Riemann invariant

$$u_n - u_{pipe} = -\frac{2 c_{pipe}}{\gamma - 1} \left[\frac{p_n}{p_{pipe}}^{\frac{(\gamma-1)}{2\gamma}} - 1 \right] \tag{3.56}$$

Substituting the Riemann invariant, Equation (3.56), into Equation (3.55) after rearranging gives

$$Ax^2 + Bx + C = 0 \tag{3.57}$$

where

$$
\begin{aligned}
x &= p_n^{\frac{\gamma-1}{2\gamma}} \\
A &= \frac{\gamma}{\gamma-1}\frac{p_{pipe}^{\frac{1}{\gamma}}}{\rho_{pipe}} + 2\left(\frac{c_{pipe}}{\gamma-1}\right)^2 p_{pipe}^{\frac{1-\gamma}{\gamma}} \\
B &= -\tfrac{1}{2}\left(u_{pipe} + \frac{2}{\gamma-1}c_{pipe}\right)\frac{c_{pipe}}{\gamma-1}p_{pipe}^{\frac{1-\gamma}{2\gamma}} \\
C &= \tfrac{1}{2}\left(u_{pipe} + \frac{2}{\gamma-1}c_{pipe}\right)^2 - \left(\frac{\gamma}{(\gamma-1)}\frac{p_{pipe}}{\rho_{pipe}} + \frac{1}{2}u_{pipe}\right)
\end{aligned}
\tag{3.58}
$$

Orifice Valves

For flow between any two reservoirs the mass flow rate is given by the standard equations for flow through an orifice assuming a large area ratio so that the pressure at the effective valve throat is equal to the pressure in the down stream reservoir.

From the conservation of energy

$$
u_t = \sqrt{\frac{2\gamma}{\gamma-1}\frac{p_{res}}{\rho_{res}}\left(1-\frac{p_t}{p_{res}}^{\frac{(\gamma-1)}{\gamma}}\right)} \tag{3.59}
$$

The mass flow rate through the orifice is then given by

$$
\dot{m} = \rho_t u_t A_t
$$

$$
\dot{m} = A\rho_{res}\left(\frac{p_t}{p_{res}}\right)^{\frac{1}{\gamma}}\sqrt{\frac{2\gamma}{\gamma-1}\frac{p_{res}}{\rho_{res}}\left(1-\frac{p_t}{p_{res}}^{\frac{(\gamma-1)}{\gamma}}\right)} \tag{3.60}
$$

Once the orifice becomes choked the throat pressure is dependant on the stagnation conditions only.

$$
p_t = p_{res}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{3.61}
$$

The mass flow rate is then given by,

$$
\dot{m} = A\rho_{res}\sqrt{\gamma\frac{p_{res}}{\rho_{res}}}\left(\frac{2}{\gamma+1}\right)^{\frac{(\gamma+1)}{2(\gamma-1)}} \tag{3.62}
$$

Conclusion

All of the possible valve flow regimes need be included in the model for inlet and exhaust valves. A decision tree selects the appropriate equations for the valve type, and state in the reservoir and pipe, and applies the correct boundary conditions. Of all possible flow regimes fully supersonic outflow and inflow have been omitted. The transition to each of these requires a non-stationary shockwave travelling in the attached pipe. The magnitude of these shocks at the boundary, and how to specify the boundary conditions are not clear. As shock waves in inlet and exhaust manifolds are uncommon in IC engines under normal operating conditions they may be safely left out of the boundary condition treatment.

For each valve the effective throat area can be either a constant, dependant on crank angle, or in the case of a reed valve, dependant on the flow conditions at the valve. In all cases the effective throat area is dependent by the geometry of the valve. The effective valve throat area is determined by experimental flow testing.

References

[1] Benson, R.S. (1982) The Thermodynamics and Gas Dynamics of Internal-Combustion engines, Clarendon press, Oxford. Vol1.

[2] Winterbone, D.E. and Pearson. R.J. (2000) Theory of engine manifold design-Wave action methods for IC engines, SAE, Warrendale.

[3] Hirsch. C. (1990) Numerical computation of internal and external flows, John Wiley & Sons, England, Vol 2, chapter 19.

[4] Vandevoorde. M., Vierendeels. E., Dick. E. and Sierens. R. (1998) A new total variation diminishing scheme for the calculation of the one-dimensional flow in inlet and exhaust pipes of internal combustion engines, Proc Instn Mech Engrs, Vol 212 part D, Pg 437-448

4. REED VALVE DEFLECTION

Introduction

Reed valve induction in two stroke engines relies on a thin reed valve, which deflects to an open position in response to the pressure variation between the crankcase and the atmosphere. A brief overview of traditional methods used to model reed motion is given in the introduction of this chapter and the remainder of this chapter presents the use of the finite element method (FEM) to model dynamic reed deflection. When the crankcase expands, and crankcase pressure drops below atmospheric, the reed valve is forced open by the pressure difference across it allows air to flow into the crankcase. As the crankcase volume contracts, due to piston motion, and the pressure rises above atmospheric the reed is forced shut against the reed block trapping the air for induction. Figure 4-1 shows a schematic outline of a reed valve.

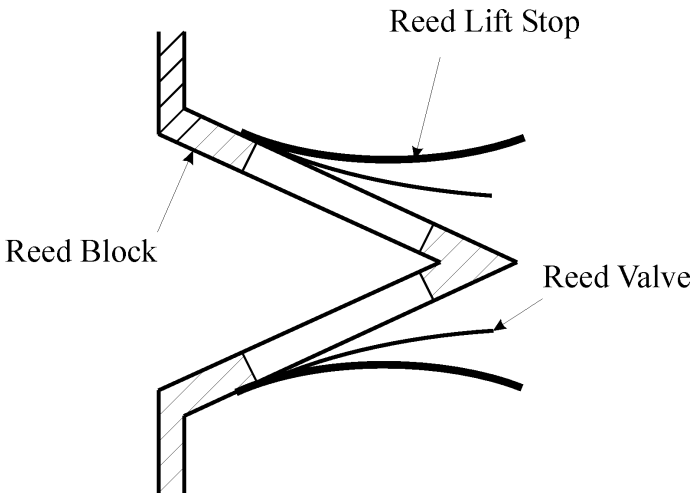


Figure 4-1. Schematic diagram of a reed valve.

The design of the reed is a balance between lightweight design to maximise the response to the input forcing function (pressure difference), and strength to withstand fatigue. A compromise is required so that the first natural frequency of the reed is above the operating frequency of the engine, while being sufficiently flexible to allow good induction. In general, this implies that design and dynamics of the reed play an important part in the response to engine cycle pressure variations, especially in high-

speed engines. A dynamic reed valve model is therefore necessary to model two-stroke reed induction engines.

The simplest approach is to treat the reed as a discrete spring mass system with the equivalent spring stiffness of the corresponding cantilevered bar (reed), as shown in Figure 4-2. The resulting dynamic equations of motion are integrated to obtain the tip deflection at each time step. For small deflections the stiffness of the reed is closely approximated. Bounce of the reed off the reed block as the reed snaps shut could also be included by a secondary spring mass system representing the reed block. Reed bounce does lead to complication however due to the non-linear nature of impact.

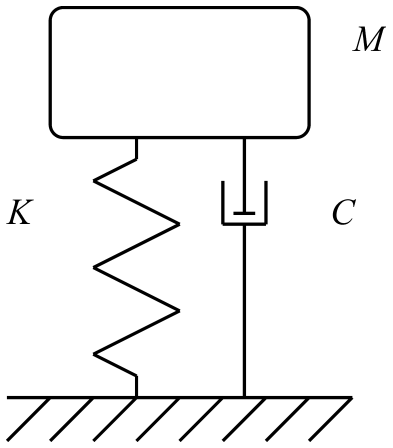


Figure 4-2. Diagram of discrete spring mass reed model (where K is the spring stiffness, M is the mass and C is the damping).

This model has only one degree of freedom and cannot predict the effect of the second mode of vibration on reed deflection. As most reed valves are designed to have their first natural frequency at least twice that of the maximum crankcase pressure fluctuation frequency, this simple approach generally does not lead to large error.

The next level of approximation in modelling is to consider the reed as a continuous beam or plate system. The first few vibration mode shapes and frequencies of the reed are calculated. The system equations are then transformed to a set of modal equations that are amenable to solution. The results are transformed back to obtain the displacement of the reed in time. A disadvantage of this technique is that all possible mode shapes, including those when the reed is in contact with the reed stop, are required before the simulation is started. The work by Hinds and Blair [1], and Fleck, Blair and Houston [2], outline this method in the context of reed valve simulation.

The finite element method was chosen to model reed valve deflection because it is a well established technique and leads to a straightforward numerical implementation. The method is accurate and leads to a general model.

Finite element method

In the finite element method the exact solution is replaced by elements, that approximate the exact function by interpolation between nodes. The interpolation functions are amenable to solution by the standard finite element method. Bernoulli-Euler beam elements where chosen to solve the reed deflection problem.

To ensure continuity of the bending moment between Bernoulli-Euler beam elements the rotation at each node along with the deflection needs to be obtained resulting in elements with four degrees of freedom. Figure 4-3 shows a beam element with the displacement and rotation at each node.

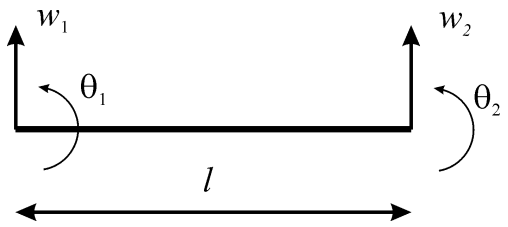


Figure 4-3. Reed element

$$\mathbf{d_e} = [w_1, \theta_1, w_2, \theta_2]^T \tag{4.1}$$

where $\mathbf{d_e}$ is the element nodal displacement vector, $w_{1,2}$ are the displacements and $\theta_{1,2}$ are the rotations at the nodes.

A commonly used choice for the interpolating function is a cubic polynomial [3].

$$w(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 \tag{4.2}$$

where $\alpha_{1,2,3,4}$ are known constants and $w(x)$ is the displacement of any point on the element.

Substituting,

$$\begin{aligned} w(0) &= w_1 \\ \frac{dw(0)}{dx} &= \theta_1 \\ w(1) &= w_2 \\ \frac{dw(1)}{dx} &= \theta_2 \end{aligned} \tag{4.3}$$

Upon rearrangement the cubic polynomial of Equation (4.2) can be re-cast as.

$$w(x) = n_1(x)w_1 + n_2(x)\theta_1 + n_3(x)w_2 + n_4(x)\theta_2 \tag{4.4}$$

The shape functions, n_1, n_2, n_3 *and* n_4, for each of the four degrees of freedom have the value of 1 at their base node and 0 at the opposite node. The shape functions are given by:

$$\begin{aligned} n_1 &= 1-3\left(\frac{x}{l}\right)^2 + 2\left(\frac{x}{l}\right)^3 \\ n_2 &= x\left[1-2\left(\frac{x}{l}\right)+\left(\frac{x}{l}\right)^2\right] \\ n_3 &= 3\left(\frac{x}{l}\right)^2 - 2\left(\frac{x}{l}\right)^3 \\ n_4 &= x\left[-\left(\frac{x}{l}\right)+\left(\frac{x}{l}\right)^2\right] \end{aligned} \tag{4.5}$$

Finally Equation (4.4) can be written as

$$u = \mathbf{N_e}\mathbf{d_e} \tag{4.6}$$

Where u is the displacement at a point on the element, $\mathbf{d_e}$ is the vector of nodal displacements and $\mathbf{N_e}$ is the vector of element shape functions

$$\mathbf{N_e} = \left[n_1(x), n_2(x), n_3(x), n_4(x)\right] \tag{4.7}$$

Stiffness matrix

It is assumed that axial tension and compression strains are negligible therefore strain energy within the reed is due to bending only. The strain energy per unit length due to beam bending is given by.

$$\text{SE/length} = \frac{EI}{2}\left(\frac{d^2u}{dx^2}\right)^2 \tag{4.8}$$

Where, *E*, is Young's modulus and, *I*, is the second moment of area for the reed element.

With suitable substitutions Equation (4.8) can be integrated to obtain the standard form for the strain energy, U_e of the element.

$$\mathbf{U_e} = \frac{1}{2}\mathbf{d^T K_e d} \tag{4.9}$$

Where the stiffness matrix, $\mathbf{K_e}$, for the Bernoulli-Euler beam element, is defined:

$$\mathbf{K_e} = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \tag{4.10}$$

where, *l*, is the length of the element.

Mass matrix

If the distributed inertial load is given by ρa, where ρ is density and a is acceleration, then integrating over a volume gives the inertial force:

$$f_I = -\int_V \rho a dV \tag{4.11}$$

where f_I is the inertial force

However, the acceleration within an element can be approximated by

$$a = \mathbf{N_e}\ddot{\mathbf{d}} \tag{4.12}$$

Where $\ddot{\mathbf{d}}$ is the vector of nodal accelerations, defined:

$$\ddot{\mathbf{d}} = \left[\frac{\partial^2 w_1}{\partial t^2}, \frac{\partial^2 \theta_1}{\partial t^2}, \frac{\partial^2 w_2}{\partial t^2}, \frac{\partial^2 \theta_2}{\partial t^2}\right] \tag{4.13}$$

Substituting Equation (4.12) into Equation (4.11) gives

$$F_i = -\int_V \rho[\mathbf{N_e}^\mathbf{T}\mathbf{N_e}]\ddot{\mathbf{d}} dV = -\mathbf{M_e}\ddot{\mathbf{d}} \tag{4.14}$$

The matrix, $\mathbf{M_e}$, is the mass matrix of the system, defined by:

$$\mathbf{M_e} = \int_V \rho \mathbf{N_e}^\mathbf{T}\mathbf{N_e} dV \tag{4.15}$$

Substituting the simple beam shape functions in Equation (4.5) into the term for, $\mathbf{M_e}$, Equation (4.15), gives.

$$\mathbf{M_e} = \left(\frac{m}{420}\right)\begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \tag{4.16}$$

Where m is the mass of the element.

The dynamic equations of motion for the complete reed system after element assembley can be written as:

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f(t)} \tag{4.17}$$

where **M** is the global mass matrix, **K** is the global stiffness matrix and **f(t)** is the input forcing function

Damping

The modelling of damping forces relies on an empirical model however they can generally be assumed to be viscous so that damping forces are proportional to nodal velocities. With this assumption Equation (4.17) becomes.

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f(t)} \tag{4.18}$$

where **C** is the global damping matrix and $\dot{\mathbf{d}}$ is the vector of nodal velocities

Rayleigh damping, or proportional damping, was selected to formulate the damping. For Rayleigh damping the damping matrix is taken to be a combination of the mass matrix, **M**, and the stiffness matrix, **K**, according to the equation.

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{4.19}$$

where α and β are free scalar parameters

The modal damping coefficients for the system are then defined:

$$\zeta_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2} \tag{4.20}$$

where ζ_i and ω_i are the modal damping ratios and rotational frequencies for the i^{th} mode respectively

The two parameters, α and β, can be set to obtain desired damping properties for two modes with the rest following Equation (4.20). As the stiffness and mass matrices for the system are already present, and the system degrees of freedom were relatively small, it is simple and effective to extract the undamped free vibration frequencies for the system using a numerical eigenvalue solver. For the reed valve the damping ratio was set to 0.01 for the fundamental mode, and 1 for the highest mode present. A small amount of damping is essential to remove transient oscillations that would otherwise persist in the solution.

Numerical Integration Algorithm

An implicit Newmark-β numerical integration scheme was selected to integrate the reed equation of motion. Consider a generic second order differential equation,

$$\mathbf{A}\mathbf{d} + \mathbf{B}\dot{\mathbf{d}} + \mathbf{C}\ddot{\mathbf{d}} = \mathbf{f(t)} \tag{4.21}$$

The Newmark-β numerical integration scheme is derived by considering Equation (4.21) at time $t = t_{n+1}$ [4]. The system in Equation (4.21) is therefore represented by:

$$\mathbf{A}\mathbf{d}_{\mathbf{n+1}} + \mathbf{B}\dot{\mathbf{d}}_{\mathbf{n+1}} + \mathbf{C}\ddot{\mathbf{d}}_{\mathbf{n+1}} = \mathbf{f}_{\mathbf{n+1}}\mathbf{(t)} \tag{4.22}$$

where $\mathbf{d_{n+1}}$, $\dot{\mathbf{d}}_{\mathbf{n+1}}$ and $\ddot{\mathbf{d}}_{\mathbf{n+1}}$ represent the unknown displacements, velocities and accelerations respectively at the time $t = t_{n+1}$

The displacements and velocities at $t = t_{n+1}$ are defined [4]:

$$\mathbf{d}_{\mathbf{n+1}} = \mathbf{d}_{\mathbf{n}} + \Delta t\,\dot{\mathbf{d}}_{\mathbf{n}} + \frac{\Delta t^2}{2}\left[\ddot{\mathbf{d}}_{\mathbf{n}}(1-\beta_1) + \ddot{\mathbf{d}}_{\mathbf{n+1}}\beta_1\right] \tag{4.23}$$

$$\dot{\mathbf{d}}_{\mathbf{n+1}} = \dot{\mathbf{d}}_{\mathbf{n}} + \Delta t\left[\ddot{\mathbf{d}}_{\mathbf{n}}(1-\beta_2) + \ddot{\mathbf{d}}_{\mathbf{n+1}}\beta_2\right] \tag{4.24}$$

where $\mathbf{d_n}$, $\dot{\mathbf{d}}_{\mathbf{n}}$ and $\ddot{\mathbf{d}}_{\mathbf{n}}$ are the known displacements, velocities and accelerations respectively at the time $t = t_n$

The value of β_1 and β_2 determine the implicit nature of the integration scheme. For $\beta_1 = \beta_2 = {}^1/_2$ the values of the second derivative of $\mathbf{d}$, at time $t = t_{n+1}$, are required to evaluate $\mathbf{d_{n+1}}$. This scheme is implicit and unconditionally stable. Conversely, for $\beta_1 = 0$, $\beta_2 = {}^1/_2$ the scheme is fully explicit and conditionally stable. Equations (4.22)-(4.24) give three equations in three unknowns that can be solved explicitly.

$$\ddot{\mathbf{d}}_{\mathbf{n+1}} = \left(\frac{\mathbf{A}\Delta t^2}{2}\beta_1 + \mathbf{B}\Delta t\beta_2 + \mathbf{C} \right)^{-1} \times \left\{ \begin{array}{l} f_{n+1} - \mathbf{A}\left[\mathbf{d}_\mathbf{n} + \Delta t\,\dot{\mathbf{d}}_\mathbf{n} + \frac{\Delta t^2}{2}\ddot{\mathbf{d}}_\mathbf{n}(1-\beta_1) \right] \\ -\mathbf{B}\left[\dot{\mathbf{d}}_\mathbf{n} + \Delta t\,\ddot{\mathbf{d}}_\mathbf{n}(1-\beta_2) \right] \end{array} \right\} \tag{4.25}$$

Once equation (4.25) has been updated at $t = t_{n+1}$ the result can be substituted back into Equation (4.23) and (4.24) to obtain $\mathbf{d_{n+1}}$ and $\dot{\mathbf{d}}_\mathbf{n}$.

For the reed valve simulation the values of, $\beta_1 = \beta_2 = ½$, were chosen to give an implicit scheme and ensure unconditional stability. The time step for the reed valve simulation could then be made to match the time step size of the global engine simulation without the need to consider the stability criterion of the reed model integration. The global engine simulation used a four step Runge-Kutta time stepping algorithm, therefore the time step size for the reed was set to match each sub-step of the global simulation.

Results

Finite element model correlation

As a check on the finite element reed model the steady state response to a sine wave input was compared with an exact analytic solution. For this purpose a generic reed valve model was used, as shown in Figure 4-4.

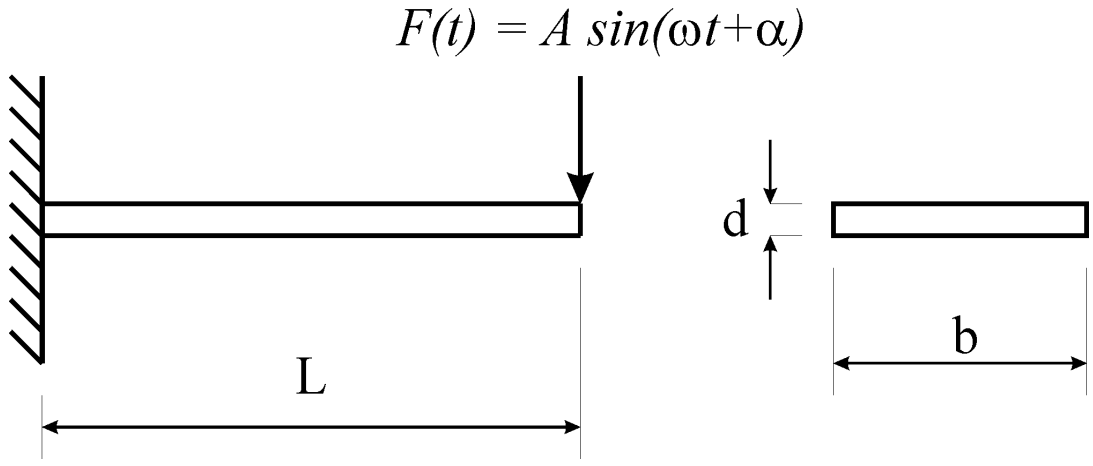


Figure 4-4. Diagram of reed petal dimensioning.

Length	L	0.03	M
Width	b	0.02	M
Thickness	d	0.0002	M
Young's Modulus	E	200	Gpa
Reed density	ρ	7800	kg/m3
Number of elements		5 (10 DOF)	
Input frequency	f	200	Hz
Min natural frequency (from FEM)	f_{min}	272.7	Hz
Max natural frequency (from FEM)	f_{max}	$1.159*10^5$	Hz
Damping ratio at ω_{min}	ξ_1	0.01	
Damping ratio at ω_{max}	ξ_2	1	
Damping coefficient	α	26.2	
Damping coefficient	β	2.7×10^6	

Table 1. Properties of generic reed valve.

Whilst the tip loading is not similar to the loading experienced by a reed, it does have an exact analytic solution against which the reed model may be correlated to validate the code. The finite element solution is started from rest and allowed to develop subject to the time varying input load. As time increases the transient response of the reed should die out and leave the steady state deflection obtained in the exact steady state case. The properties of the transient part of the solution are also useful to investigate properties of the damping matrix. The analytical solution for the steady state condition is defined:

$$u(x,t) = f(x)\sin(\omega t + \alpha) \tag{4.26}$$

where u is lateral displacement and,

$$f(x) = H_1 \cosh(\frac{\mu x}{L}) + H_2 \sinh(\frac{\mu x}{L}) + H_3 \cos(\frac{\mu x}{L}) + H_4 \cosh(\frac{\mu x}{L}) \tag{4.27}$$

For a thin beam the equation of motion is:

$$EI \frac{\partial^4 u}{\partial x^4} + \rho \frac{\partial^2 u}{\partial t^2} = 0 \tag{4.28}$$

Substituting the analytical solution in Equation (4.27), into Equation (4.28), gives the term for μ in Equation (4.27):

$$\mu = L \sqrt[4]{\frac{\rho \omega^2}{EI}} \tag{4.29}$$

By applying boundary conditions at each end of the beam the constants, H_n, may be obtained. The boundary conditions for this case are:

1) At $x = 0$, $f = 0$
2) At $x = 0$, $\frac{\partial f}{\partial x} = 0$
3) At $x = L$, $F(t) = -EI \frac{\partial^3 u}{\partial x^3}$, (Equate load to the shear force at the end of the beam)
4) At $x = L$, $\frac{\partial^2 u}{\partial x^2} = 0$, (zero bending moment at the end of the beam)

These boundary conditions lead to the following values for the constants, H_n, in Equation (4.27).

$$H_1 = \frac{-\frac{A}{EI}\left(\frac{L}{\mu}\right)^3}{\left(\sinh(\mu) - \sin(\mu)\right) - \frac{\left(\cosh(\mu) + \cos(\mu)\right)^2}{\left(\sinh(\mu) + \sin(\mu)\right)}}$$

$$H_2 = -H_1 \frac{\left(\cosh(\mu) + \cos(\mu)\right)}{\left(\sinh(\mu) + \sin(\mu)\right)}$$

$$H_1 = -H_3$$

$$H_2 = H_4 \tag{4.30}$$

Discussion

Finite element simulation for the reed valve shows good agreement with an exact steady state, periodic solution. For a small number of elements, 5, with 10 degrees of freedom (DOF), the finite element method is an excellent candidate for reed valve modelling. From the limited amount of validation the method has been shown to be computationally efficient and accurate.

For comparison with the analytic case, an input window was used over the first five cycles to avoid having a discontinuous initial condition. The input window is simply a multiplier that continuously changes from zero to one over the first few periods of the input to ensure the initial conditions for the problem are continuous and well posed. Without an input window the input function is discontinuous in its first derivative, which causes excitation of multiple vibration modes. Figure 4-5 shows the reed response with the addition of a window function applied to the input. The finite element solution shows close correlation with the exact solution.

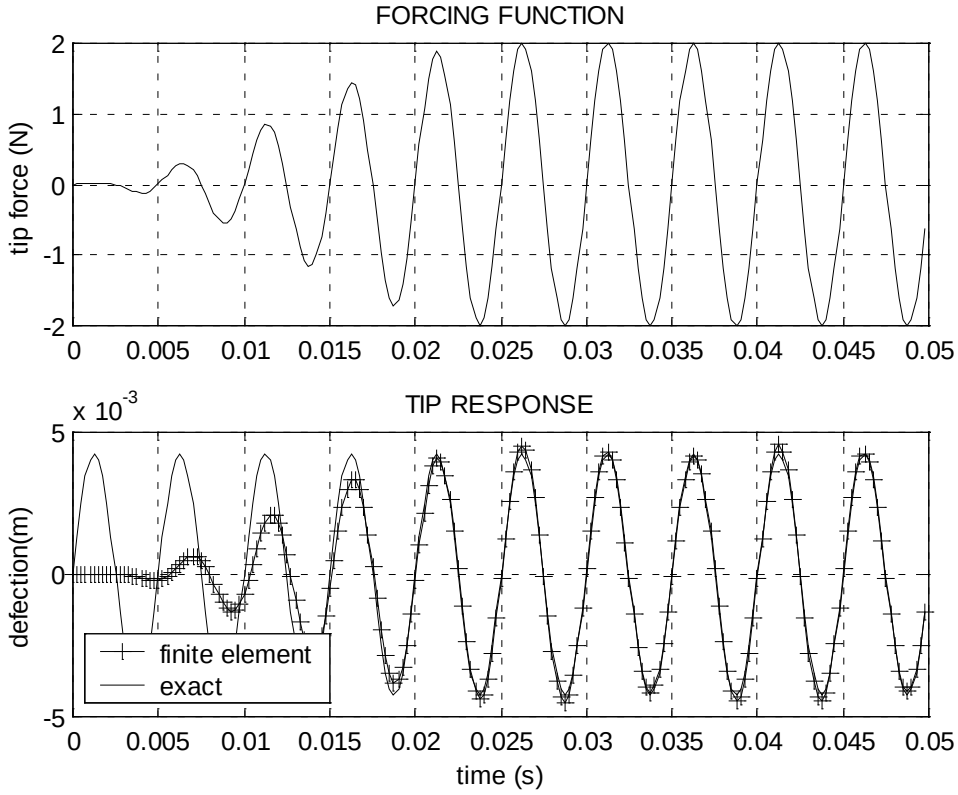


Figure 4-5. Reed response with no damping and windowed input function.

Figure 4-6 highlights the effect of not having an input window. The first derivative of the input function is discontinuous at time, $t = 0$, exciting higher order modes of vibration in the reed. These are gradually damped out by the damping terms.

In the absence of any damping higher order modes are excited and without any means to dissipate the energy associated with these modes they persist in the solution. Clearly persistent spurious oscillations are undesirable and damping is essential. It would be expected that aerodynamic damping would be dominant. However, Rayleigh damping does not model aerodynamic loads. Aerodynamic damping would be included as part of the time dependant pressure load input function. One-dimensional gas dynamics simulation, from which the pressure boundary condition for the reed are obtained, are incapable of modelling this type of damping. Hence, application of aerodynamic damping terms remains an avenue for future investigation.

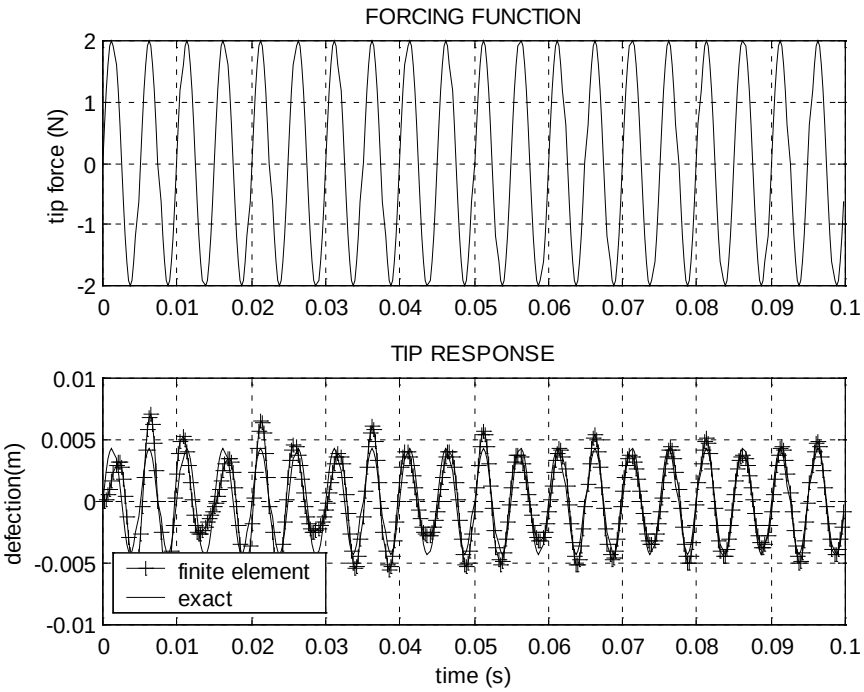


Figure 4-6. Reed response with damping, no input window.

For most two stroke, reed valve engines, bounce of the reed off the lift stop, and reed block, play a significant part in the overall reed dynamics. The current scheme does not include reed bounce but could be extended to model it. The lift stop would add an additional nodal force when the reed is in contact with the stop. Modelling the impulsive forces as the reed hits the lift stop results in significant complication of the scheme due to non-linear impact.

The general literature suggests both transverse and longitudinal vibrational modes influence the reed bounce event [5]. To adequately capture the physics in this situation the FEM would need to be extended to a two dimensional case. This approach would also be a natural option for three dimensional air flow/reed petal interaction research.

The finite element method appears to be an excellent method for reed valve simulation. The reed model has been trouble free within the engine simulation code. The accuracy of the valve lift could benefit from further validation. The natural extension to two dimensions would increase the DOF however not excessively. If 12 quadrilateral plate elements, each with 12 DOF, were used to model the reed, the whole system would have 48 DOF, remaining computationally efficient and suitable for engine simulation.

References

[5] Hinds. E. T. and Blair. G. P. (1978) *Unsteady Gas Flow Through Reed Valve Induction Systems*, SAE paper 780766.

[6] Fleck. R., Blair. G. P. and Houston. R. A. (1987) *An Improved Model for Predicting Reed Valve Behaviour in Two-Stroke Cycle Engines*, SAE paper 871654.

[7] Astley, R. J. (1992) *Finite elements in solids and structures*, Chapman and Hall, London.

[8] Zienkiewicz, O.C. and Taylor, R. L (1990) *The Finite Element Method*, 4th edition, McGraw-Hill, London, Vol 2, chapter 10.

[9] Matianec, W. and Bogusz. A. (1996) *Theoretical and experimental study of gas flow through reed valve in a two-stroke engine*, SAE paper 961802 (In: *Two-stroke engine technology and emissions*, SAE).

5. THERMODYNAMICS

Governing equation

The computation of combustion incorporates a single zone thermodynamic first law analysis. This method assumes a homogeneous mixture throughout the cylinder. Combustion is modelled with a heat release mechanism calculated by an empirical formula. Figure 5-1 shows a schematic representation of a variable volume open thermodynamic system representing a cylinder.

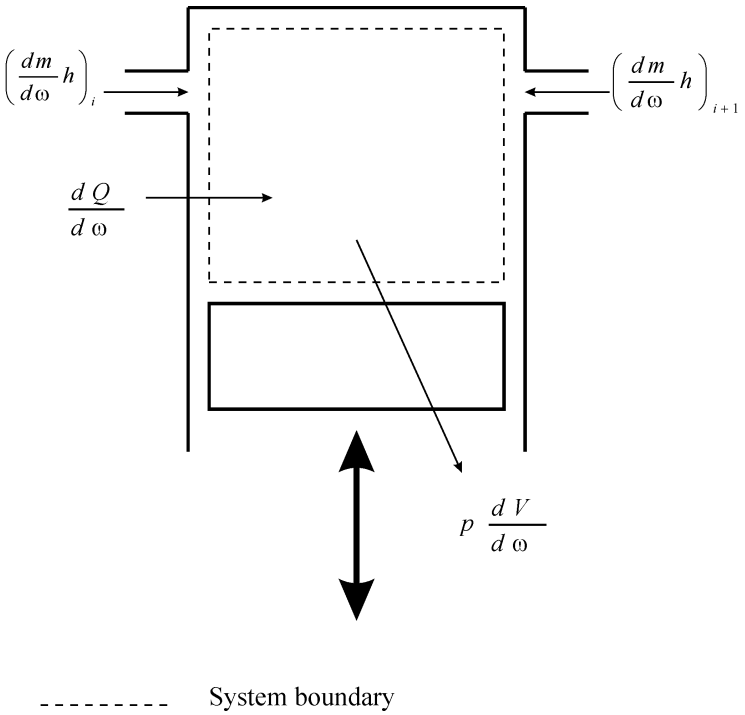


Figure 5-1. Schematic diagram for thermodynamic analysis of a cylinder system.

Considering the external exchange of energy across the system boundaries.

$$\frac{dU}{d\omega} = \frac{dQ}{d\omega} - p\frac{dV}{d\omega} + \sum_i \frac{dm_i}{d\omega} h_i \tag{5.1}$$

where U = internal energy of cylinder contents, Q = heat transfer, P = pressure, V = volume, m = mass, h = specific enthalpy and ω = crank angle.

The last term in Equation (5.1) represents the separate mass flows across the system boundary. The change of energy in the system is also expressed by relating the change of internal state within the volume, leading to:

$$\frac{dU}{d\omega} = m\frac{du}{d\omega} + u\frac{dm}{d\omega} \tag{5.2}$$

where u = specific internal energy

To a first order approximation Equation (5.2) can be rewritten:

$$\frac{dU}{d\omega} = mC_v(T)\frac{dT}{d\omega} + \left(u_p(T) - u_r(T)\right)\frac{dm}{d\omega}_{burn} + u_{cyl}(T)\sum_i \frac{dm}{d\omega}_i \tag{5.3}$$

where C_v = mean specific heat capacity at constant volume, T = temperature, u_p = total internal energy of products, u_r = total internal energy of reactants and u_{cyl} is the mean internal energy of fluid in the cylinder

The second term in Equation (5.3) represents the energy released due to combustion. Combining Equations (5.1) and (5.3) leads to an expression for the rate of change of temperature of the cylinder contents.

$$\frac{dT}{d\omega} = \frac{\frac{dQ}{d\omega} - p\frac{dV}{d\omega} + \sum_i \frac{dm_i}{d\omega}h_i - \left(u_p^\circ(T) + u_r^\circ(T)\right)\frac{dm}{d\omega}_{burn} - u_{cyl}(T)\sum_i \frac{dm}{d\omega}_i}{mC_v(T)} \tag{5.4}$$

where $\frac{dm}{d\omega}_{burn}$ is the mass burn rate of combustion

Equation (5.4) gives an expression for the rate of change of temperature with crank angle in the cylinder as a function of volume change, mass flows across the system boundary, heat transfer from the system and combustion. During the crank angle period when combustion does not occur the burn rate is set to zero.

For a small time step the temperature at the new time can be approximated by.

$$T_{n+1} = T_n + \Delta\omega\frac{dT}{d\omega} \tag{5.5}$$

where the subscript n denotes a discrete point in time

Similarly the mass of each species in the cylinder can be updated from the mass flows across the system boundary and the rate of combustion by:

$$m_{n+1}^{j} = m_{n}^{j} + \Delta\omega\left(\sum_{i}\frac{dm^{j}}{d\omega}_{i} + \frac{dm_{burn}^{j}}{d\omega}\right) \tag{5.6}$$

where the subscript *j* denotes a particular chemical species and *i* denotes separate mass flows across the system boundary

The formation of combustion products is approximated by assuming formation at chemical equilibrium. Once the reactants become products it is assumed no further chemical reactions occur.

Once the cylinder mass, temperature and volume have been updated for the new time step, cylinder pressure follows from the ideal gas law.

$$p_{n+1} = \left(\rho RT\right)_{n+1} \tag{5.7}$$

where ρ = density and R = gas constant

Combustion heat release model

In a simple single zone first law combustion analysis, combustion is modelled as heat release into the cylinder. The empirical Wiebe function has been used to calculate the progress of combustion in spark ignition internal combustion (IC) engines. Wiebe combustion models rely on an empirical curve relating mass fraction burned to crank angle. The shape of the mass fraction profile is determined by a curve fit to actual cylinder pressure data, Heywood [1].

$$x_{burn} = 1 - \exp\left[-a\left(\frac{\omega - \omega_{\circ}}{\Delta\omega}\right)^{m+1}\right] \tag{5.8}$$

Where ω is the crank angle, ω_0 is the onset of combustion, Δw is the combustion duration and x_{burn} is the mass fraction burnt. The parameters ‘a’ and ‘m’ are used to fit the curve to a particular engine. Heywood [1] gives the values, a = 5 and m = 2, as indicative for a typical IC engine. Figure 5-2 shows the corresponding Weibe combustion ‘s’ curve

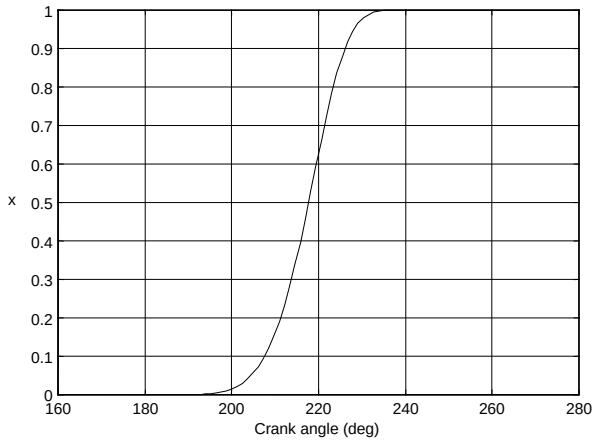


Figure 5-2. Calculated mass fraction burn for typical Wiebe 'S' curve.

Kinematics

Slider crank kinematics

The slider-crank mechanism is the standard arrangement for traditional reciprocating IC engines, as shown in figure 5-3. The kinematics for the slider-crank mechanism is easily determined by completing the standard kinematic loop closure equations for the mechanism.

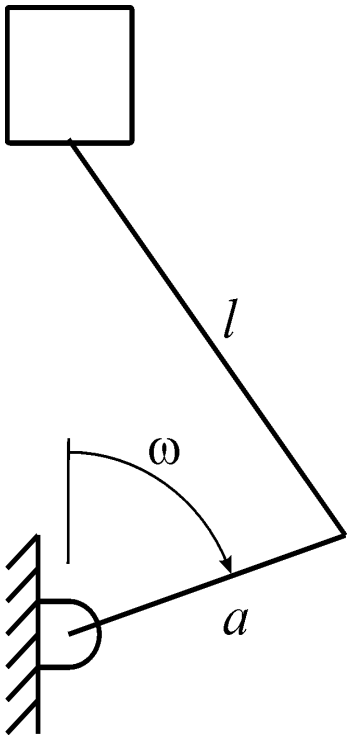


Figure 5-3. Slider crank mechanism.

By applying loop closure the position of the piston is obtained from which the volume of the cylinder as a function of crank angle can be calculated.

Applying this method yields:

$$\frac{V}{V_c} = 1 + \tfrac{1}{2}(CR - 1)\left[R + 1 - \cos w - \left(R^2 - \sin^2 \omega\right)^{1/2}\right] \tag{5.9}$$

where

$$R = \frac{l}{a}$$

l = connecting rod length

a = crank throw

V_c = clearance volume

V = cylinder volume

CR = compression ratio

ω = crank angle

Pivotal kinematics

The kinematics of the pivotal engine, shown in figure 5-4, is based on the four-bar linkage. An approach similar to the slider-crank is taken to solve the pivot piston angle as a function of crank angle. The loop closure relations are applied that yield a 2 by 2 non-linear system of equations. These were then solved by the Newton-Raphson numerical technique.

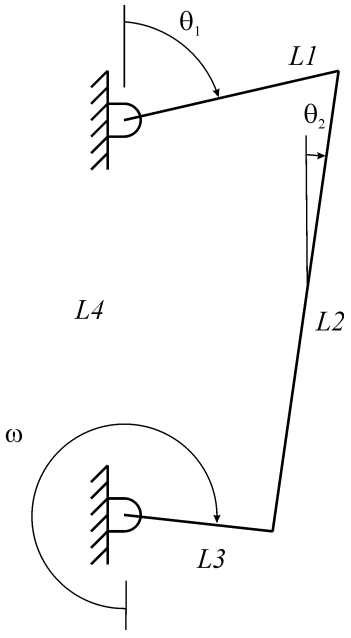


Figure 5-4. Schematic diagram of the four bar linkage.

Summing the displacements of each link in the x and y directions yields:

$$f_x = L_1 \sin\theta_1 - L_2 \sin\theta_2 + L_3 \sin(n\omega) = 0 \tag{5.10}$$

$$f_y = L_1 \cos\theta_1 - L_2 \cos\theta_2 + L_3 \sin(n\omega) + L_4 = 0 \tag{5.11}$$

The variable n has been added to allow clockwise (n = 1) and anti-clockwise (n = -1) crank rotation. Incorporating the direction of rotation in this way leads to a numerical implementation compatible with the rest of the simulation, which requires positive crank angles. The non-linear system of equations is solved from an initial guess using Newton-Raphson iterative refinement of the form.

$$\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}_{i+1} = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}_i + \begin{bmatrix}\Delta\theta_1 \\ \Delta\theta_2\end{bmatrix} \tag{5.12}$$

where Equation (5.12) is iterated until convergence. $\Delta\theta_{1,2}$ are obtained from the solution to:

$$\begin{bmatrix}\frac{\partial f_x}{\partial\theta_1} & \frac{\partial f_x}{\partial\theta_1} \\ \frac{\partial f_y}{\partial\theta_2} & \frac{\partial f_y}{\partial\theta_2}\end{bmatrix}\begin{bmatrix}\Delta\theta_1 \\ \Delta\theta_2\end{bmatrix} = \begin{bmatrix}-f_x \\ -f_y\end{bmatrix} \tag{5.13}$$

Once the angle of the pivot piston has been found the volume and surface area of the cylinder are easily determined. Figure 5-5 shows a typical relationship between the angle of the pivot piston and the angle of the crank.

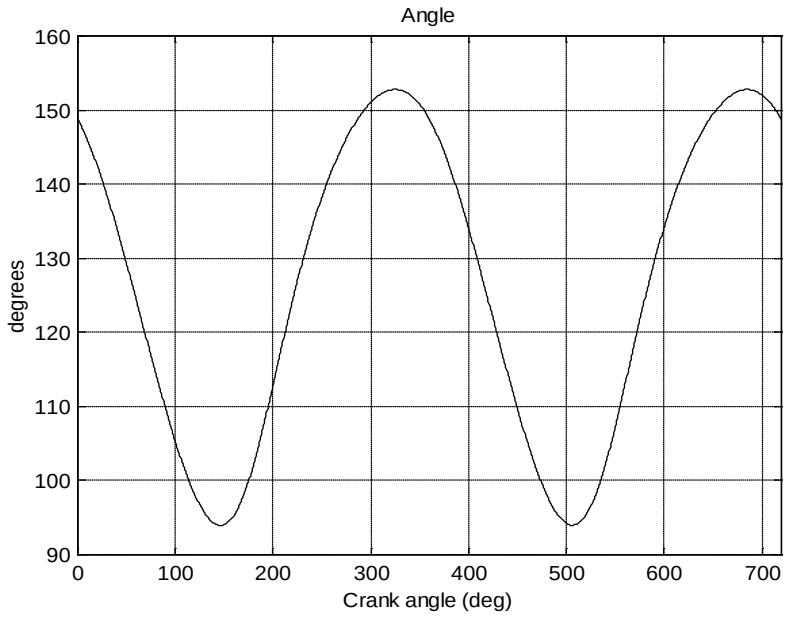


Figure 5-5. Pivot piston - crank angle relationship of a generic pivotal engine four bar linkage.

Thermochemistry

The internal energy and specific heat capacity of the 12 chemical species were calculated using the gas phase heat capacity Shomate equation. The coefficients for this equation are available online from the NIST Chemistry Webbook [2] and have been reproduced here for the relevant species in table 5-1.

The Shomate equation is given by:

$$C_p = A + B\overline{T} + C\overline{T}^2 + D\overline{T}^3 + \frac{E}{\overline{T}^2} \tag{5.14}$$

$$h^\circ - h^\circ_{298.15} = A\overline{T} + B\frac{\overline{T}^2}{2} + C\frac{\overline{T}^3}{3} + D\frac{\overline{T}^4}{4} - \frac{E}{\overline{T}} + F - \Delta_f h^\circ_{298} \tag{5.15}$$

C_p = heat capacity (J/mol×K)

h° = standard enthalpy (kJ/mol)

$\Delta_f h°_{298.15}$ = enthalpy of formation at 298.15 K (kJ/mol)

S° = standard entropy (J/mol×K)

$\overline{T}$ = temperature (K) / 1000.

To obtain the internal energy for any species, including chemical potential energy, equation (5.14) is manipulated to give.

$$e = \frac{\left(At + B\frac{\overline{T}^2}{2} + C\frac{\overline{T}^3}{3} + D\frac{\overline{T}^4}{4} - \frac{E}{\overline{T}} + F\right)}{R\overline{T}} \tag{5.16}$$

e = internal energy (kJ/mol)

Species	Range (t)	A	B	C	D	E	F	$\Delta_f h_{298}$
H	0.3 - 6	20.78603D0	4.850638D-10	-1.582916D-10	1.525102D-11	3.196347D-11	211.8020D0	217.9994D0
O	0.3 - 6	21.1861D0	-.502314D0	0.168694D0	-0.00896163D0	0.0756643D0	243.131D0	249.1731D0
N	0.3 - 6	2.11358D+01	-3.88842D-01	4.35446D-02	2.46853D-02	-2.56784D-02	4.66311D+02	4.726832D+02
H_2	0.3 - 1.5	3.31078D+01	-1.15080D+01	1.16093D+01	-2.84440D+00	-1.59665D-01	-9.99197D+00	0.000000D+00
H_2	1.5 - 6	3.41434D+01	5.03927D-01	3.72036D-01	-3.85985D-02	-8.07476D+00	-2.12188D+01	0.000000D+00
OH	0.3 - 1.3	3.22777D+01	-1.13629D+01	1.36055D+01	-3.84649D+00	-1.33540D-03	2.97511D+01	3.898706D+01
OH	1.3 -6	2.87470D+01	4.71449D+00	-8.14725D-01	5.47481D-02	-2.74783D+00	2.64144D+01	3.898706D+01
CO	0.3 - 1.3	2.55676D+01	6.09613D+00	4.05466D+00	-2.67130D+00	1.31021D-01	-1.18009D+02	-1.105271D+02
CO	1.3 - 6	3.51507D+01	1.30009D+00	-2.05921D-01	1.35503D-02	-3.28278D+00	-1.27838D+02	-1.105271D+02
NO	0.3 - 1.2	2.38349D+01	1.25888D+01	-1.13901D+00	-1.49746D+00	2.14194D-01	8.33578D+01	9.029114D+01
NO	1.2 - 6	3.59917D+01	9.57170D-01	-1.48032D-01	9.97382D-03	-3.00409D+00	7.31079D+01	9.029114D+01
O_2	0.3 - 6	2.96590D+01	6.13726D+00	-1.18652D+00	9.57803D-02	-2.19663D-01	-9.86139D+00	0.000000D+00
H_2O	0.3 - 1.7	3.00920D+01	6.83251D+00	6.79344D+00	-2.53448D+00	8.21386D-02	-2.50881D+02	-2.418264D+02
H_2O	1.7 - 6	4.19643D+01	8.62205D+00	-1.49978D+00	9.81194D-02	-1.11576D+01	-2.72180D+02	-2.418264D+02
CO_2	0.3 - 1.2	2.49974D+01	5.51870D+01	-3.36914D+01	7.94839D+00	-1.36638D-01	-4.03608D+02	-3.935224D+02
CO_2	1.2 - 6	5.81664D+01	2.72007D+00	-4.92289D-01	3.88435D-02	-6.44729D+00	-4.25919D+02	-3.935224D+02

Table 5-1. Shomate equation coefficients for relevant chemical species, sourced from the NIST chemistry Webook [2].

Internal energy of fuel

The fuel was assumed to have a mean composition given by.

$C_aH_bO_cN_d$

The total internal enthalpy of the fuel was then approximated by the polynomial expression.

$$h_{fuel} = At + B\frac{t^2}{2} + C\frac{t^3}{3} + D\frac{t^4}{4} - \frac{E}{t} + F \tag{5.17}$$

The coefficients for various fuels have been given by Heywood [1]. Typically for simulations herein Octane, C_8H_{16}, has been used as the fuel. The coefficients given below for C_8H_{16} have been sourced from Heywood, converted to give h_{fuel} in kilo-Joules per mole with a 298.15 Kelvin datum.

A = -2.3143

B = 759.8981

C = -409.1408

D = 85.3620

E = -0.1295

F = -254.182

Equilibrium Chemistry

Typically [8] in first law analysis of IC engines the major chemical species may be assumed to be in chemical equilibrium. The time scales of the major reactions, with the notable exception of NOx, are sufficiently less than that of combustion. Assuming shifting equilibrium therefore gives an accurate prediction of chemical species for combustion calculations. Where NO_x formation is of interest a kinetic chemistry model for these species may be used in conjunction with the equilibrium model. Olikara and Borman [3] gave a method for calculating the equilibrium chemistry that has been widely used in engine simulation and is the same method presented here.

Reaction equation

The combustion of fuel with air and residual gases may be modelled by the chemical equation.

$$z\left[\begin{matrix} xr_{13}\left(C_a H_b O_c N_d\right) \\ +\, xr_1 H + xr_2 O + xr_3 N + xr_4 H_2 + xr_5 OH + xr_6 CO + xr_7 NO + xr_8 O_2 + xr_9 H_2O + xr_{10} CO_2 + xr_{11} N_2 + xr_{12} Ar \end{matrix}\right] \rightarrow$$

$$x_1 H + x_2 O + x_3 N + x_4 H_2 + x_5 OH + x_6 CO + x_7 NO + x_8 O_2 + x_9 H_2O + x_{10} CO_2 + x_{11} N_2 + x_{12} Ar$$

(5.18)

Where $x_{1\text{-}12}$ are the mole fractions of the products, $xr_{1\text{-}13}$ are the mole fractions of the reactants, including the fuel, and z is the amount of reactants that give one mole of products.

Main reactions

Typically combustion chemistry is very complex, and almost always unknown, however Olikara and Borman [3] have proposed 7 simplified hypothetical reactions that represent combustion in IC engines.

Chemical reaction	Equilibrium coefficient	
$\frac{1}{2} H_2 \Leftrightarrow H$	$K_1 = \frac{x_1 p^{\frac{1}{2}}}{x_4^{\frac{1}{2}}}$	(5.19)
$\frac{1}{2} O_2 \Leftrightarrow O$	$K_2 = \frac{x_2 p^{\frac{1}{2}}}{x_8^{\frac{1}{2}}}$	(5.20)
$\frac{1}{2} N_2 \Leftrightarrow N$	$K_3 = \frac{x_3 p^{\frac{1}{2}}}{x_{11}^{\frac{1}{2}}}$	(5.21)
$\frac{1}{2} H_2 + \frac{1}{2} O_2 \Leftrightarrow OH$	$K_5 = \frac{x_5}{x_4^{\frac{1}{2}} x_8^{\frac{1}{2}}}$	(5.22)
$\frac{1}{2} O_2 + \frac{1}{2} N_2 \Leftrightarrow NO$	$K_7 = \frac{x_7}{x_8^{\frac{1}{2}} x_{11}^{\frac{1}{2}}}$	(5.23)
$H_2 + \frac{1}{2} O_2 \Leftrightarrow H_2O$	$K_9 = \frac{x_9}{x_4 x_8^{\frac{1}{2}} p^{\frac{1}{2}}}$	(5.24)

$$CO + \tfrac{1}{2}O_2 \Leftrightarrow CO_2 \qquad K_{10} = \frac{x_{10}}{x_6 x_8^{\frac{1}{2}} p^{\frac{1}{2}}} \tag{5.25}$$

where p = pressure

Rearranging the terms for the equilibrium constants.

$$x_1 = C_1 x_4^{\frac{1}{2}} \qquad C_1 = \frac{K_1}{p^{\frac{1}{2}}} \tag{5.26}$$

$$x_2 = C_2 x_8^{\frac{1}{2}} \qquad C_2 = \frac{K_2}{p^{\frac{1}{2}}} \tag{5.27}$$

$$x_3 = C_3 x_{11}^{\frac{1}{2}} \qquad C_3 = \frac{K_3}{p^{\frac{1}{2}}} \tag{5.28}$$

$$x_5 = C_5 x_4^{\frac{1}{2}} x_8^{\frac{1}{2}} \qquad C_5 = K_5 \tag{5.29}$$

$$x_7 = C_7 x_8^{\frac{1}{2}} x_{11}^{\frac{1}{2}} \qquad C_7 = K_7 \tag{5.30}$$

$$x_9 = C_9 x_4 x_8^{\frac{1}{2}} \qquad C_9 = K_9 p^{\frac{1}{2}} \tag{5.31}$$

$$x_{10} = C_{10} x_6 x_8^{\frac{1}{2}} \qquad C_{10} = K_{10} p^{\frac{1}{2}} \tag{5.32}$$

Atomic balance

Performing a balance on the number of atoms present in the reaction, equation (5.18).

C balance:

$$x_6 + x_{10} = \left[axr_{13} + xr_6 + xr_{10}\right]z \tag{5.33}$$

H balance:

$$x_1 + 2x_4 + x_5 + 2x_9 = \left[bxr_{13} + xr_1 + 2xr_4 + xr_5 + 2xr_9\right]z \tag{5.34}$$

O balance

$$\begin{aligned} &x_2 + x_5 + x_6 + x_7 + 2x_8 + x_9 + 2x_{10} = \\ &\left[cxr_{13} + xr_2 + xr_5 + xr_6 + xr_7 + 2xr_8 + xr_9 + 2xr_{10}\right]z \end{aligned} \tag{5.35}$$

N balance:

$$x_3 + x_7 + 2x_{11} = \left[dxr_{13} + xr_3 + xr_7 + 2xr_{11}\right]z \tag{5.36}$$

Ar balance:

$$x_{12} = [xr_{12}]z \tag{5.37}$$

Finally the sum of the mole fractions must be unity giving.

$$\sum_{i=1}^{12} x_i = 1 \tag{5.38}$$

Equation system formulation

A 4 by 4 non-linear system of equations can now be formulated by eliminating all terms except x_4, x_6, x_8, and x_{11}.

From the C and H balance.

$$C_1 x_4^{\frac{1}{2}} + 2x_4 + C_5 x_4^{\frac{1}{2}} x_8^{\frac{1}{2}} + 2C_9 x_4 x_8^{\frac{1}{2}} - \frac{r_2}{r_1}\left(x_6 + C_{10} x_6 x_8^{\frac{1}{2}}\right) = 0 \tag{5.39}$$

$$r_1 = axr_{13} + xr_6 + xr_{10} \tag{5.40}$$

$$r_2 = bxr_{13} + xr_1 + 2xr_4 + xr_5 + 2xr_9 \tag{5.41}$$

From the O balance

$$C_2 x_8^{\frac{1}{2}} + C_5 x_4^{\frac{1}{2}} x_8^{\frac{1}{2}} + x_6 + C_7 x_8^{\frac{1}{2}} x_{11}^{\frac{1}{2}} + 2x_8 + 2C_9 x_4 x_8^{\frac{1}{2}} + 2C_{10} x_6 x_8^{\frac{1}{2}} - \frac{r_3}{r_1}\left(x_6 + C_{10} x_6 x_8^{\frac{1}{2}}\right) = 0 \tag{5.42}$$

$$r_3 = cxr_{13} + xr_2 + xr_5 + xr_6 + xr_7 + 2xr_8 + xr_9 + 2xr_{10} \tag{5.43}$$

From N balance

$$C_3 x_{11}^{\frac{1}{2}} + C_7 x_8^{\frac{1}{2}} x_{11}^{\frac{1}{2}} + 2x_{11} - \frac{r_4}{r_1}\left(x_6 + C_{10} x_6 x_8^{\frac{1}{2}}\right) = 0 \tag{5.44}$$

$$r_4 = dxr_{13} + xr_3 + xr_7 + 2xr_{1110} \tag{5.45}$$

From the sum of mass fractions and Ar balance

$$\begin{aligned} &C_1 x_4^{\frac{1}{2}} + C_2 x_8^{\frac{1}{2}} + C_3 x_{11}^{\frac{1}{2}} + x_4 + C_5 x_4^{\frac{1}{2}} x_8^{\frac{1}{2}} + x_6 + C_7 x_8^{\frac{1}{2}} x_{11}^{\frac{1}{2}} + x_8 + C_9 x_4 x_8^{\frac{1}{2}} \\ &+ C_{10} x_6 x_8^{\frac{1}{2}} + x_{11} - \frac{r_5}{r_1}\left(x_6 + C_{10} x_6 x_8^{\frac{1}{2}}\right) - 1 = 0 \end{aligned} \tag{5.46}$$

$$r_5 = xr_{12} \tag{5.47}$$

Equations (5.39), (5.42), (5.44) and (5.46) form a non-linear system of equations in four variables that can be solved by the standard Newton-Raphson numerical technique (e.g.

Griffiths and Smith [4]), to give the equilibrium concentrations of the combustion products.

Initial estimate

To start the Newton-Raphson solution for the 4 by 4 non-linear system an initial guess of the mole fractions is required. This must be sufficiently close to the sought solution otherwise convergence will not be achieved. A method to calculate an initial estimate has been given by Olikara and Borman [3]. It involves reducing the number of chemical species involved and using the solution for the simplified chemical equilibrium as an initial estimate in the larger reaction model. Assuming the chemical products are H_2, CO, H_20, CO_2 and Ar only the chemical equation is modelled by.

$$z\left[r_1C + r_2H + r_3O + r_4N + r_5Ar\right] \rightarrow \\ x_4H_2 + x_6CO + x_9H_2O + x_{10}CO_2 + x_{12}Ar \tag{5.48}$$

Now a similar technique is applied to the reduced set of equations.

C balance

$$x_6 = \frac{zr_1}{1 + C_{10}x_8^{\frac{1}{2}}} \tag{5.49}$$

H balance

$$x_4 = \frac{1}{2}\left(\frac{zr_2}{1 + C_9x_8^{\frac{1}{2}}}\right) \tag{5.50}$$

N balance

$$zr_4 = 2x_{11} \tag{5.51}$$

O balance

$$zr_3 = x_6 + 2x_8 + x_9 + 2x_{10} \tag{5.52}$$

Ar balance

$$zr_5 = x_{12} \tag{5.53}$$

The terms $r_{1\text{-}5}$ remain unchanged from the larger system. Combining terms gives an equation for the reduced chemical equilibrium model, in a single variable, that can be solved by the Newton-Raphson method.

$$r_1 \frac{(1+2C_{10}x_8^{\frac{1}{2}})}{(1+C_{10}x_8^{\frac{1}{2}})} + \frac{1}{2} r_2 \frac{C_9 x_8^{\frac{1}{2}}}{(1+C_9 x_8^{\frac{1}{2}})} + \frac{2x_8}{z} - r_3 = 0 \tag{5.54}$$

Equation (5.54) requires an estimate for *z* that is obtained from the condition that the sum of the products must add to unity leading to.

$$z = \frac{1}{r_1 + r_2 + r_3 + r_4 + r_5} \tag{5.55}$$

Evaluating equilibrium constants

The equilibrium constants, used to calculate the combustion chemistry, were approximated by the method given in the Appendix of Olikara and Borman [3]. Olikara and Borman used an empirical equation that gave the equilibrium constant for each reaction as a function of temperature in the 600K to 4000K range. The JANAF [5] thermochemical tables provided the source data. The empirical equation was of the form.

$$\log(K_p) = A\ln T + \frac{B}{T} + C + DT + ET^2 \tag{5.56}$$

Where T is the temperature in Kelvin.

Table 5-2 lists the coefficients found by curve fit to the JANAF [5] tables. Olikara and Borman [3] originally gave the coefficients for a transformed temperature scale however the values given here are for equation (5.56) in absolute temperature (Kelvin).

Reaction	*A*	*B*	*C*	*D*	*E*
$\frac{1}{2}H_2 \Leftrightarrow H$	4.3216E-01	-1.1246E+04	-3.1256E-01	-7.4569E-05	2.4244E-09
$\frac{1}{2}O_2 \Leftrightarrow O$	-8.4898E-01	-1.3478E+04	8.9552E+00	6.3770E-04	-5.8312E-08
$\frac{1}{2}N_2 \Leftrightarrow N$	3.8970E-01	-2.4583E+04	4.5312E-01	-9.6362E-05	5.8555E-09
$\frac{1}{2}H_2 + \frac{1}{2}O_2 \Leftrightarrow OH$	-1.4179E-01	-2.1331E+03	1.8329E+00	3.5502E-05	-3.1024E-09
$\frac{1}{2}O_2 + \frac{1}{2}N_2 \Leftrightarrow NO$	1.5084E-02	-4.7096E+03	5.4190E-01	2.7302E-06	-1.5446E-09
$H_2 + \frac{1}{2}O_2 \Leftrightarrow H_2O$	-7.5235E-01	1.2421E+04	2.5942E+00	2.5955E-04	-1.6268E-08
$CO + \frac{1}{2}O_2 \Leftrightarrow CO_2$	5.8301E-02	1.4891E+04	-5.1533E+00	8.6329E-05	-5.6680E-09

Table 5-2. Coefficients for the estimation of the equilibrium constants calculated by curve fit to the source data in JANAF [5].

Heat Transfer

Heat transfer from the cylinder has been calculated using a spatially averaged instantaneous heat transfer coefficient calculated by the method due to Annaud [1]. The method treats the cylinder as a short pipe and applies non-dimensional analysis to arrive at an empirical equation for the heat transfer coefficient with parameters that are determined from experiment.

For a typical slider crank engine Annaud proposed:

$$Nu = 0.35\,\mathrm{Re}^{0.7} \tag{5.57}$$

Nu = Nusselt number

Re = Reynolds's number

The Reynolds's number of the cylinder is calculated using a mean piston velocity, S and the cylinder bore, B.

$$\mathrm{Re} = \frac{\rho S B}{\mu} \tag{5.58}$$

For a slider crank engine the mean piston velocity is given by.

$$S = 2 \times stroke \times N \tag{5.59}$$

where N is the rotational speed of the engine. The mean heat transfer coefficient is then given by

$$h = \frac{Nu \times k}{B} \tag{5.60}$$

h = heat transfer coefficient

k = thermal conductivity

B = cylinder bore

For these calculations it was necessary to have appropriate values for the thermal conductivity and viscosity, each of which vary with temperature and composition of the gas. These were approximated by the polynomial expressions given in Blair [6].

$$k = 0.0061944 + 7.3814 \times 10^{-5} T - 1.2491 \times T^2 \quad \text{(W/mK)} \tag{5.61}$$

$$\mu = 7.457 + 4.1547 \times 10^{-6} T - 7.4793 \times 10^{-8} T^2 \quad \text{(kg/ms)} \tag{5.62}$$

The Annaud model has been extended to the Pivotal engine by assuming an equivalent bore for the Pivotal engine and a mean pivot piston velocity. This assumption is not necessarily entirely accurate. In the absence of heat transfer data from the Pivotal engine with which to establish a more accurate empirical model it has been assumed it gives a reasonable first approximation of heat transfer. As the kinematics of the four-bar linkage result in a non-linear system of equations, estimating the mean pivot piston velocity presents some difficulty, without resorting to numerical techniques. It has therefore been assumed that the formula used for the slider-crank, will give a reasonable result.

$$S = 2 \times stroke \times N$$

Psychrometric's

The moisture content of air at varying relative humidity is required to determine the properties of air. For this purpose the saturation pressure of water in air has been fitted to a polynomial as a function of temperature.

$$P_{sat} = 1 \times 10^3 \left(a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4 \right) \tag{5.63}$$

$a_5 = 9.23814 \times 10^{-7}$

$a_4 = -1.03533 \times 10^{-3}$

$a_3 = 4.38107 \times 10^{-1}$

$a_2 = -82.8845$

$a_1 = 5.91039 \times 10^{-3}$

Where T is the temperature in Kelvin and P_{sat} is the saturation pressure of water in air (Pa). The coefficients, $a_{1\text{-}5}$, have been determined by a least squares curve fit to source data from the Rogers and Mayhew [7] steam tables. Once the Saturation pressure has been determined the partial pressure of water in air is given by.

$$p_{H20} = p_{sat} \times \phi \tag{5.64}$$

Where ϕ = relative humidity, and p_{H20} is the partial pressure of saturated water in air. The moles of water per mole of Oxygen in air is then given by.

$$QH_2O = (1 + QN_2 + QCO_2 + QAr) \frac{PH_2O}{(Patm - PH_2O)} \tag{5.65}$$

QH_2O = moles H_20 per mole O_2 in air.

QN_2 = moles N_2 per mole O_2 in air

QCO_2 = moles CO_2 per mole O_2 in air

QAr = moles AR per mole O_2 in air

Scavenge

A scavenge model determines the concentration of residual gases in the exhaust flow during cylinder scavenge. Ideally the cylinder residuals and the fresh charge will not mix and the exhaust flow will contain displaced residual gases only. At the other theoretical extreme the inflow can be considered to short circuit the cylinder and immediately exit the cylinder through the exhaust. In-between these extremes the cylinder can be considered as perfectly mixing. This assumes there is no spatial variation of concentration in the cylinder and any charge that enters the cylinder instantly forms a uniform mix with the cylinder contents. In reality scavenge can be considered as a combination of all three.

Various empirical scavenge models, based on theoretical considerations, have been proposed that include parameters to fit the model to a particular engine. Benson [8],

and Blair [6] give an overview of these models. The general assumptions required for these models are that scavenge occurs at constant pressure and volume. Clearly these assumptions cannot be proved. Blair [6] has proposed the following empirical scavenge model based on theoretical considerations:

$$SE_V = 1 - e^{\left(\kappa_0 + \kappa_1 SR_V + \kappa_2 SR_V^2\right)} \tag{5.66}$$

$$SR_V = \frac{V_{as}}{V_{cyl}}$$

$$SE_V = \frac{V_{ta}}{V_{cyl}}$$

Where, V_{as} is the volume of air supplied, V_{ta} is the volume of air trapped and V_{cyl} is the volume of the cylinder. The parameters κ_0, κ_1, κ_2 are included to fit the equation to a particular engine.

An alternative model for scavenge could be to consider the residency time distribution function for the cylinder, Nauman and Buffham [9]. The concentration of residuals in the exhaust would then be of the form,

$$C_{out}(\theta) = \int_{-\infty}^{\theta} C_{in}(\theta') f(\theta - \theta') d\theta' \tag{5.67}$$

Where C_{out} is the concentration of the out flowing gases, C_{in} is the concentration of the inflowing gases, θ is the time co-ordinate and $f(\theta)$ is the residence time distribution function for the cylinder. An estimate of $f(\theta)$ from experimental or theoretical considerations could then be made to calculate the concentration of the exhaust flow leaving the cylinder.

Currently only a perfect mixing model has been implemented in the engine simulation program. Perfect mixing is adequate for 4-stroke engine simulations because valve overlap is comparatively small in the 4-stroke cycle but underestimates the exhaust residuals in a typical 2-stroke engine. Development of a more advance scavenge model has been left for future development.

Discussion

The single zone cylinder model is adequate for the evaluation of power output and general performance but is incapable of predicting pollutant formation. This is due to a lack of a kinetic chemistry model to predict NO_x formation. Implementation of kinetic chemistry would be best done in conjunction with a two-zone model, due to the more accurate representation of temperature variation in the cylinder. In a two-zone model the cylinder is divided into an unburnt and burnt zone, separated by a flame front. Using the separate zones, the temperature in the burnt zone is more accurately calculated. This results in improved kinetic chemistry calculations due to their dependence on the cylinder temperature.

The form of the single zone thermodynamic model presented here does not account for chemical reaction within the burnt products after combustion. The form of first law equations developed here would require post combustion reactions to be modelled by the complete kinetic chemistry of the 7 representative chemical equations and addition NOx equations. Benson [8] in his single zone model, calculated the chemistry of the burnt products as shifting equilibrium. This produces satisfactory results but leads to complication in deriving the thermodynamic model due to the dependence of the gas constant on the shifting equilibrium. Additional terms in equation (5.2) are required to account for the internal energy variation due to shifting equilibrium. The approach adopted here would allow chemical rate based reactions by treating reaction like a virtual mass flow, with reactant species removed and product added into the system. An additional heat release term could account for the heat of formation of the reaction.

Figure 5-6 shows a comparison between the simulation described in this thesis, called Puttputt, and Ricardo Wave, a commercial engine simulation tool. The figure plots the cylinder pressure against cylinder volume on logarithmic scales for a conventional single cylinder four-stroke engine (see example in the appendix). The two simulations show good agreement except on the expansion stroke. The greater slope during expansion shown in the Puttputt simulation is due to the lack of a combustion chemistry model in the burnt products of the Puttputt model. The value for the ratio of specific heats during expansion is underestimated leading to a more negative slope than Ricardo Wave, which calculates chemical reaction in the burnt zone as shifting equilibrium. Development of a post combustion chemistry model, using a rate base combustion model, has been left for future development.

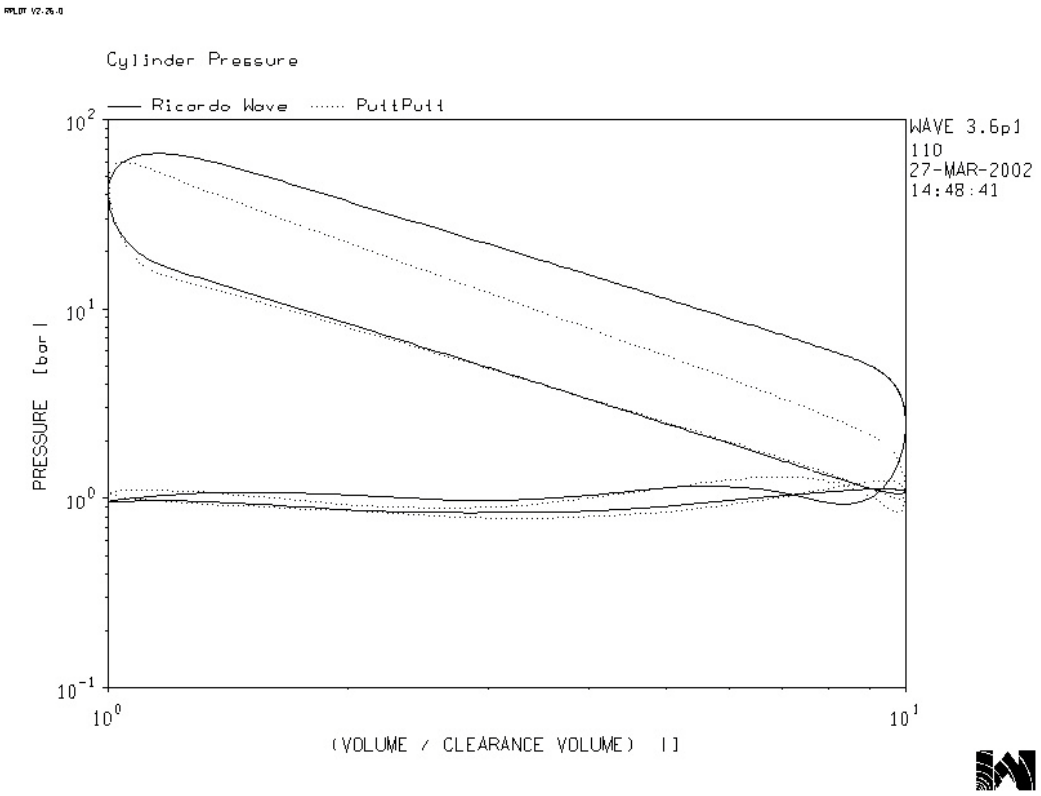


Figure 5-6. Comparison of a conventional single cylinder four stroke engine at 3000 rpm simulated on Ricardo Wave 3.6, a commercial engine simulation tool, and the Puttputt engine simulation program described in this thesis. The input file for this engine has been given in the appendix.

To calculate the cylinder pressure at a new time step it has been assumed that the cylinder contents are an ideal gas. The assumption is reasonable for the burnt products, but the fuel in the fresh charge will certainly not behave ideally. This has been overlooked in the simulation, and fuel has been treated as an ideal gas regardless. This has been partially excused by the fact that fuel is only a small proportion of the fresh charge. The non-ideal behaviour of the fuel therefore has only a small effect on the overall accuracy.

References

[1] Heywood, J. B. (1988) *Internal combustion engine fundamentals*, Page 678, McGraw-Hill, New York.

[2] NIST (2001) *Chemistry WebBook*, http://webbook.nist.gov/chemistry/

[3] Olikara, C. and Borman, G. L. (1975) *A computer program for calculating properties of equilibrium combustion products with some applications to I.C engines*, SAE paper 750468.

[4] Griffiths, D. V. ands Smith, I. M. (1991) *Numerical methods for engineers*, Blackwell Scientific, Oxford.

[5] JANAF (1971) *JANAF thermochemical tables*, 2nd edition, The Dow Chemical CO, Midland, Michigan.

[6] Blair, G. P. (1996) *Design and simulation of two-stroke engines*, SAE, Warrendale.

[7] Rogers, G. F. C. and Mayhew, Y. R. (1995) *Thermodynamics and transport properties of fluids*, 5th edition, Blackwell, Cambridge U.S.A.

[8] Benson, R. S. (1982) *The Thermodynamics and Gas Dynamics of Internal-Combustion engines*, Clarendon press, Oxford. Vol 1 and 2.

[9] Nauman, E. B. and Buffham, B. A. (1983) *Mixing in continuous flow systems*, John Wiley & Sons, New York.

6. OUTLINE OF FORTRAN CODE

The simulation project has been written in Fortran 95 using Compaq Visual Fortran. The simulation code has been written in modules so that each module simulates a discrete function of an engine. The modules collectively joined together form the building blocks of the simulation tool. Communication between modules is achieved via a common boundary data type and data transfer protocols that are common to all modules.

Simulation modules were written so that they were effectively self-contained simulations of the individual parts concerned. Each module needs no communication with other parts of the simulation, except through its boundary condition data type, and to determine time step size. All other operations and instructions for the module were written in self-contained collections of functions and subroutines within the module. Figure 6-1 outlines the flow of execution for the simulation program. Time step size is determined by the minimum CFL condition of all the gas dynamics components. This was found to be the limiting time step size and determines the time step for all simulation components.

The connectivity of components is not defined in the code therefore the arrangement needs to be determined from the input file and made at run time. The connectivity between components is defined by a numbering system that assigns a unique integer identifier to connected components. This numbering system is included as part of the input file and the connection is made at run time by pointer target association between the common boundary data types of each module. More detail on the numbering system and setting up of the input file can be found in the appendix: Using The Simulation Tool, which also has example input files. The flexibility in the way that components can be arranged provides significant scope in the application of the simulation tool to a diverse range of engines and gas dynamics simulations.

The simulation program is written in a modular format. Each simulation component has its own data structure and subroutines that perform operations exclusively on the variables in their own data structure. The individual program components are effectively self-contained. The extensive use of data types provides a methodical way to organise the code although it does however fall some way short of object oriented

programming (OOP). Implementation of the simulation program in an object orientated language may prove beneficial to future development of the simulation.

Program flow diagram outline.

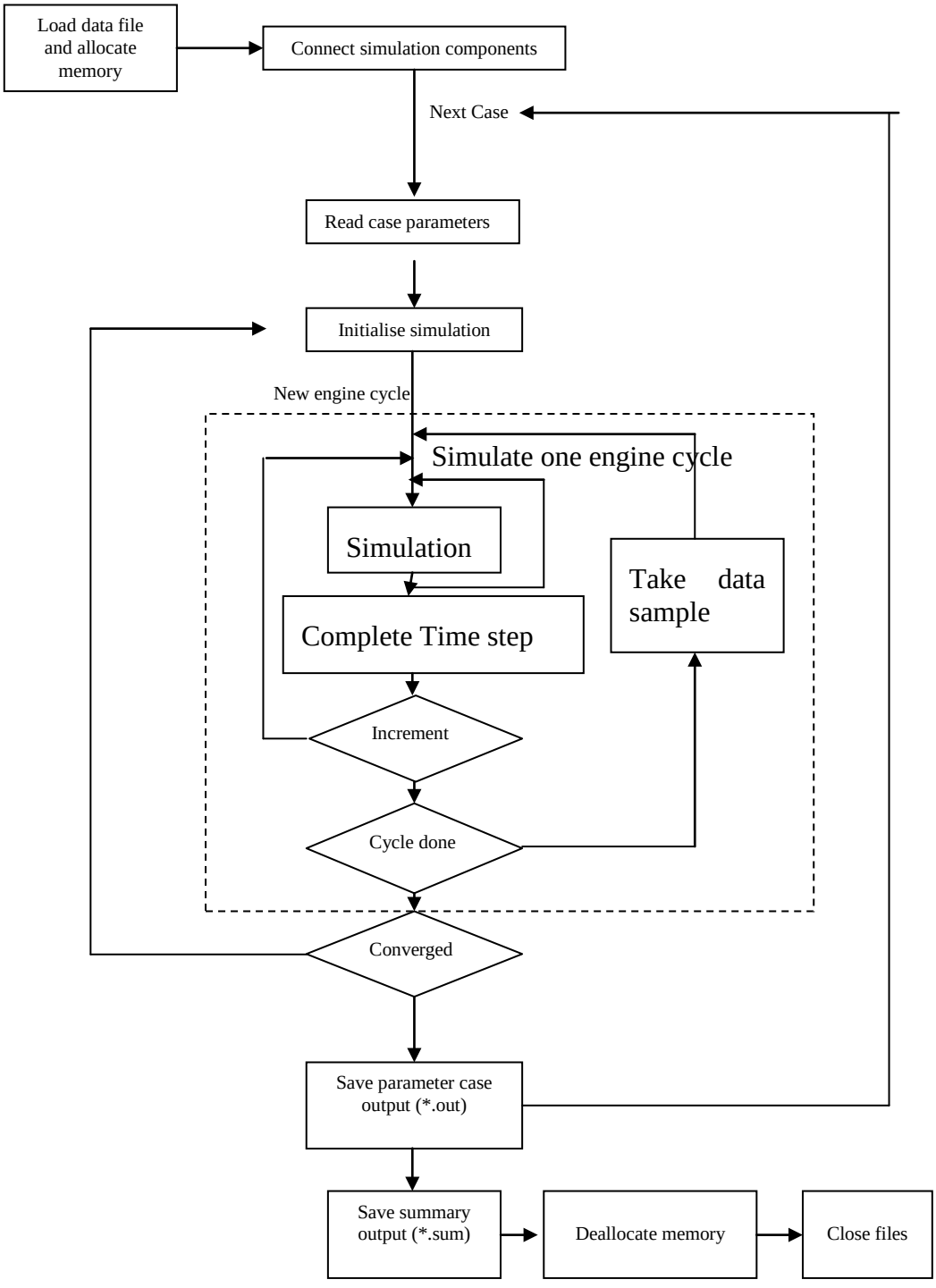


Figure 6-1. Flow diagram of Fortran code execution.

Load data file

The input data text file is opened from the *Engine* subroutine. Each component type then has its input data loaded into their data types.

Connect simulation components

Each component has a common state and flux data type within its data structure with integer IDs, read in from the input file, that determines to which nodes the component connects. Pointer target association is then used to connect components with common nodes. At this point the *CheckConnectivity* subroutine is also called to check that all components have their flux and state data types associated. If this test fails an error message is output and the program halts.

Read case parameters

The parameter matrix allows input as a variable rather than a numerical value. A call to 'ReadParameter' matrix is made to update these parameter variables for the current parameter case.

Initialise simulation

The initial conditions for all components are set. All components are reset to their initial conditions at the beginning of each case. Future development of the code may have this as an option to allow for faster convergence when re-initialisation is not essential.

Simulation sub-step

Inner loop of Runge-Kutta time stepping scheme of the form.

$$\overline{p}^{j+1} = p_i + RK^j \Delta t \frac{\partial \overline{p}^j}{\partial t}$$

$$j = 1,2,3,4$$

$$RK = (\tfrac{1}{4}, \tfrac{1}{3}, \tfrac{1}{2}, 1)$$

Time step size is determined by minimum CFL condition for all pipes present in the simulation.

Complete time step

Once the sub-steps are complete all variables are updated to the new time.

$$p_{i+1} = \overline{p}^4$$

Increment

The simulation is forced to pass through regular crank increments to take samples of the solution for output. If a crank increment is reached a call to *TakeDataSample* is made.

Take data sample

Cycles through each component. If the Boolean output variable for the component is set to 'true' the component places it's current state into its output data structure.

Cycle done

If the end of a thermodynamic cycle is reached the code proceeds to check for convergence, otherwise it continues with the current thermodynamic cycle.

Converged

Compares the output power from the previous and current cycle. If the relative change is less than specified by the input file, or the maximum number of cycles has been reached, the simulation case is considered complete. The program loops to the next case until all cases are done.

Save parameter case

Opens a Matlab output file (*.out) and writes detailed information about the state of each component around a thermodynamic cycle for the current parameter case. Proceeds to next case until all cases have been simulated.

Save summary

Opens a text output file (*.sum) and outputs a summary of engine performance for each parameter case.

Deallocate memory

Frees any memory dynamically allocated for the simulation components.

Close files

Closes all output files and exits simulation

7. DISCUSSION

The simulation project has been a success in that an engine simulation software program has been successfully developed, however the project did not reach the goal of validating the code against real engine data. The program is a useful tool but needs to be validated before its predictions can be used with confidence. Sample output from the simulation has been given in the appendix (Using The Simulation Tool), which includes various graphs for the generic pivotal engine example. Engine simulation is a relatively mature field of research therefore the techniques used in the simulation project represent up to date methods that have been developed and refined by many researchers over the last 3 decades.

Gas dynamics

The gas dynamics model is well developed and represents current methodologies used in computational fluid dynamics. Validation against engine pipe flows is still required although the gas dynamics code has given good agreement against analytic test cases. These test cases have been used to check spatial accuracy, time accuracy, shock resolution, heat transfer and pipe friction. In each case the numerical fluid dynamics solution has compared well indicating that the gas dynamics code has accurately captured the physics of gas flow in pipes.

The most obvious improvement or extension would be to include the variation of specific heat capacity of each species in exhaust pipe flows, which have significant temperature gradients. The pipe simulation already calculates the concentration of each species along a pipe. Therefore, implementation of variable specific heat capacity calculations would be relatively straightforward.

Valve boundary conditions

The implementation of boundary conditions for pipe flow incorporates a rigorous treatment of the characteristic information at the pipe boundary. Conservation of mass momentum and energy is maintained at a boundary between two pipes, however only

mass is strictly conserved at the valve boundary. The multidimensional shape of the valve and port is not modelled therefore the correct prediction of valve flow rates relies on the appropriate selection of the effective valve throat area. The effective area must be calculated by experiment and is typically deduced from steady state flow experiments on the valve or port. Valve flow is typically unsteady however the steady state simplification gives a good first approximation and is a widely accepted method.

The modelling of reed valve motion in engine simulation has traditionally been done by modal analysis techniques of beam or plate continuum systems. The application of the finite element method (FEM) is relatively new in reed valve simulation, consequently the reed valve model presented is relatively simple and has opportunities for improvement. In particular the FEM model could be extended to a 2-D model and non-linear reed impact and bounce included.

Thermodynamics

A standard single zone model has been implemented to calculate the cylinder thermodynamics. The single zone analysis is the simplest method with which to analyse a cylinder and has been widely adopted. The method described in this thesis gives sufficient accuracy for performance prediction applications. The model does not include chemical kinetics therefore is incapable of predicting NO_x pollutant formation. Inclusion of kinetic chemistry would be a natural extension and should be implemented in conjunction with a multi-zone combustion model that more accurately represent the spatial temperature variation across a cylinder. The current model does not account for chemical reaction in the burnt products of combustion.

The modelling of the scavenge process in the cylinder during the valve overlap period is currently a simple fully mixed model. Perfect mixing is adequate in 4-stroke simulations however is less accurate in 2-stroke models. The development of a more advanced scavenge model should be one of the first areas for the attention of future work.

Validation

Presently the simulation is capable of running arbitrary 2-stroke engine arrangements including the Pivotal engine. The code is un-validated against any real engine data. Where possible, sub-models have been correlated against special test cases that have an analytic solution. Figure 7-1 shows the indicated power output for a conventional

single cylinder four-stroke engine (see example in the appendix) simulated with the program presented in this thesis, called Puttputt, compared with the same engine simulated on Ricardo Wave, a commercial engine simulation tool. The two predictions have reasonable agreement and exhibit the same trends in indicated power.

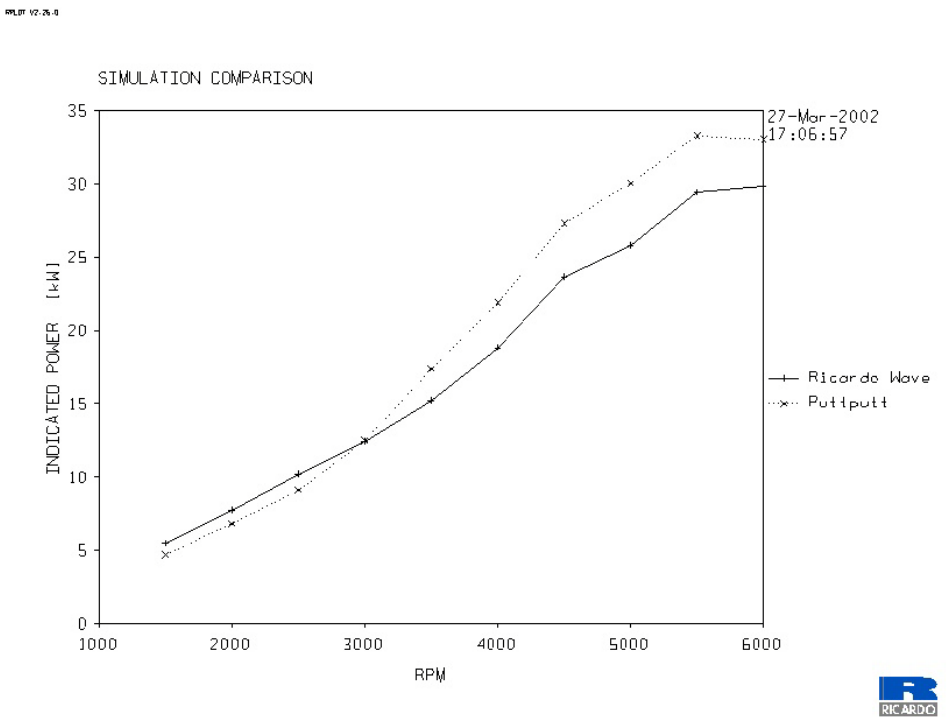


Figure 7-1. Comparison with a conventional single cylinder four stroke engine simulated on Ricardo Wave 3.6, a commercial engine simulation software, and the Puttputt engine simulation program described in this thesis. The input file for this engine has been given in the appendix.

The next logical extension beyond the current project is to validate and compare the engine simulation against a range of real engine test data, requiring the collection and analysis of extensive data from an engine.

Validation would have two main aspects:

(1). Validate to check the predictive physical models are correctly modelling the physical system they represent.

- Compare and correlate numerical and experimental data. This would include comparing each sub-model as well as the overall engine performance predictions.

(2). Calibration of empirical models against experimental data.

- Pivotal cylinder heat transfers correlation.
- Mechanical friction correlation for the Pivotal engine.
- Weibe combustion model.
- Valve flow coefficients.

Full validation of the code will require extensive data on the specification and design of the engine as well as experimental engine test data.

The range of data required for thorough validation includes:

- Full technical specification, design and dimensioning of the engine.
- Engine heat loss data across the full range of engine operation to perform an energy balance on the engine. Heat loss data is particularly required to validate the cylinder heat transfer model.
- In-cylinder pressure data in a fired engine to allow calculation of the coefficients in the Wiebe combustion model and calculation of the indicated power. The Wiebe coefficients can change with engine speed therefore in-cylinder pressure measurements would be required across the full range of engine speeds.
- Crankcase pressure data to calculate pumping losses.
- Ambient atmospheric conditions during testing.
- Air and fuel mass flow rates into the engine.
- Pressure loss data for flow through valves for the calculation of valve flow coefficients and valve effective areas.
- Flywheel power output over the engine operating range.
- Mean exhaust gas temperature for cylinder energy balance.
- Pressure at various points in the manifold systems to monitor and compare the progress of dynamic pressure waves.
- Pressure variation in the inlet and exhaust ports.

Experimental validation of the scavenge model and measurement of dynamic reed motion are more difficult to directly measure and would require specialised test apparatus or detailed numerical simulation.

Future Directions

Once validation of the computer code has been established there are some obvious areas in which the code could be extended and improved.

These include:

(1). Extension of the modelling capabilities

- Multi-zone combustion model.

- Kinetic post-combustion chemistry.
- Development of an empirical, Pivotal engine, heat transfer correlation.
- Variable heat capacities in exhaust pipe flow.
- Extension of the reed valve model to include reed bounce and 2-D effects.
- Development of a more sophisticated cylinder scavenge flow model.
- Parameter and engine design optimisation routines.

(2). Ease of use.

- Develop user interface.
- Online help
- Computationally more efficient code.

8. CONCLUSION

A general one-dimensional internal combustion (IC) engine simulation program for the Pivotal two-stroke engine has been developed. The program numerical simulates the physics occurring within an engine and where necessary uses empirical methods to complete the engine model. The code is un-validated against any real engine and validation will be required before the simulation can be used with confidence.

Engine simulation is a relatively mature field of research so that the simulation described herein reflects well developed and refined methods. The engine cylinder simulation is a simple single zone model, which has been widely used and is adequate for the evaluation of power output and general performance.

The code has been written in Fortran using Compaq Visual Fortran 6.5. It has become apparent however that the nature of the simulation problem better lends itself to the object orientated approach to programming, and is a recommended alternative for future implementation. The program has been devised in a modular and flexible way so that the single simulation tool is capable of simulating a broad range of engines.

The gas dynamics code has produced accurate results free from spurious oscillations about flow discontinuities and maintains 2^{nd} order spatial accuracy in regions of smooth flow. Pipe boundary conditions satisfy the propagation of characteristic information in the pipe so that the implementation of pipe boundary conditions remains well posed. All possible conditions at the pipe boundary have been incorporated into the boundary condition model except the case of fully super-sonic flow in the pipe to which the valve is attached. The transition to this type of flow requires non-stationary shock wave propagation at the boundary, which is difficult to calculate. As shock waves in inlet and exhaust manifolds of IC engines are uncommon and undesirable, their treatment has been omitted from the boundary condition calculations.

Deflection of the reed valve in two-stroke engines has been carried out by direct integration of the finite element formulation for the reed. The method has been computationally efficient and correlates well with an analytical deflection problem. A small amount of dampening was added to the integration scheme to dissipate the energy associated with high order modes of vibration of the reed, which would otherwise

persist in the solution. The reed model does not include bounce of the reed off the lift stop and reed block.

Tasks that have not been accomplished by the project include: complete validation of the numerical simulation against real engine data, extension of existing modelling capabilities to more advanced methods and development of an easy and efficient user interface. Should subsequent development be carried out on the simulation program, specific recommendations have been made in the discussion chapter of this thesis

Appendix: USING The SIMULATION tool

The engine simulation code is a DOS console application that is run from a DOS prompt. The application command takes two command line arguments. To run a simulation a command of the following form is typed at the command prompt.

C:\exepath\puttputt.exe c:\inputpath\inputfile.dat option

exepath Path to the directory containing the engine simulation program 'puttputt.exe'.

puttputt.exe The engine simulation program.

inputpath Directory path to the input text file 'input.dat'.

inputfile.dat The ASCII (*.dat) text input file.

option Either the text string 'test' or 'sim', 'sim' starts a simulation while 'test' causes the program to run through the input file to test for valid input without performing any calculations.

A windows dialog application is also available (on the attached disk), called 'Engine simulation.exe', that can be used to run a batch of engine simulation files.

The simulation source code is also included on the accompanying compact disk in a Compaq Visual Fortran 6.5 workspace project called *Puttputt.dsw*.

Input files

A simulation is put together by arranging and connecting simulation elements; pipes, valves, cylinders etc, to form an engine model. Each element connects to others following certain rules as described in the following sections. Provided these rules are followed almost any engine or pipe flow simulation can be devised. The simulation elements are separated into two categories, elements and virtual elements. Elements have physical dimensions e.g. pipe elements, whereas virtual elements occupy a point, eg valve elements. Once the engine has been setup the numerical values for selected engine variables can be either input directly or put into a matrix called the parameter case matrix. Each row of the matrix represents a variable, and each column a parameter

case. The program performs a simulation for each column (case) of the parameter matrix and each specified variable takes on the value in its row of the matrix for that case. Almost any variable can be specified in the parameter matrix. The parameter matrix allows the effect of varying any engine dimensions, or operating condition, to be easily assessed.

OUTLINE OF SIMULATION SETUP PROCEDURE

(1). Collect all data about the engine being simulated. The amount a data required can be comprehensive. The more accurate and complete the data is for the engine the more accurate the simulation will be.

(2). Draw a schematic diagram of the engine using the simulation elements described in this section (see examples).

(3). Give each node in the diagram a unique integer identification number.

(4). Assign a unique name to each simulation element.

(5). Write the data blocks for each element filling in all input fields.

(6). Run the simulation with the 'test' command line argument to check for valid input data.

(7). Run simulation. Parameter case output file (*.out) can be analysed using Matlab. The summary output file (*.sum) can be opened by any text editor, or read into an Excel spreadsheet for analysis.

Simulation elements

The input block for each simulation element has been given in this section. The input blocks make up the input file for a simulation. The use and purpose of each element and the way in which they can connect to other elements are outlined.

ParameterMatrix

- Required input block
- Must contain #rpm parameter variable

The parameter matrix allows variables to be a defined as a parameter rather than input as a numerical value. The first two integer values instruct the code as to the number of parameter variables, and number of parameter cases respectively. For the example ParameterMatrix input block the parameter variables #rpm and #dia1 can be used in place of a numerical value in the input file. The code will then do a simulation run for each column (parameter case) of the input ParameterMatrix. The respective numerical values of the parameter variables will take on the respective values in the columns of the input matrix. ParameterMatrix is a required input block and as a minimum must have the #rpm parameter variable.

INPUT BOCK

```
!************************************************************
ParameterMatrix
2
5

#rpm    1000    1500    2000    2500    3000
#dia1   0.05    0.05    0.05    0.05    0.05

!************************************************************
```

Setup

- Required input block
- Input for simulation options, engine type and operating conditions.

INPUT BLOCK

```
!************************************************************
Setup

CFL = 0.2
soln_tol = 1D-4
MaxCycles = #maxc
spatial_acu = 2
MaxSimTime = 30
KinematicsType = FB
NumStrokes = 2

!************************************************************
```

VARIABLES

CFL: Courant Friedrich Lewy number used by gas dynamics code to determine time step size.

soln_tol: Condition of convergence for the simulation defined by the relative change in indicated power output.

MaxCycles: Maximum number of engine cycles to allow if convergence has not already been satisfied.

spatial_acu: Specifies the spatial accuracy of the gas dynamics code. Must either be '1' or '2' for 1st and 2nd order accuracy respectively.

MaxSimTime: Maximum allowable simulation time in seconds. If the simulation time is exceeded the simulation will stop at the end of it's current engine cycle.

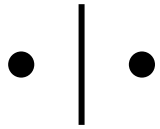
KinematicsType: Must be 'SC', 'FB' or 'none' for slider-crank, four-bar (Pivotal) or no engine kinematics respectively.

NumStrokes: Must be either '2' or '4' for two-stroke or four-stroke respectively.

Junctions

- Joins two pipe elements.
- Virtual element.
- Optional depending on engine configuration.

DIAGRAM



INPUT BOCK

```
!************************************************************
Junctions
2

UpID    DownID          Dia     name
120     121             #dia1   Junc1
122     123             #dia2   Junc2

!************************************************************
```

VARIABLES

First line after junctions specifies how many junction elements to read. Each row of the input data represents an individual junction element.

UpID: Integer identifier for the up stream pipe boundary i.e. connecting pipe outlet.

DownID: Integer identifier for the down stream pipe boundary i.e. connecting pipe inlet.

Dia: Diameter of the junction. Diameter of the junction, up stream and down stream boundary must all be equal.

Name: Name of junction as used by data output file.

PivotalKinematics

- Defined the kinematics of the pivotal engine.

DIAGRAM

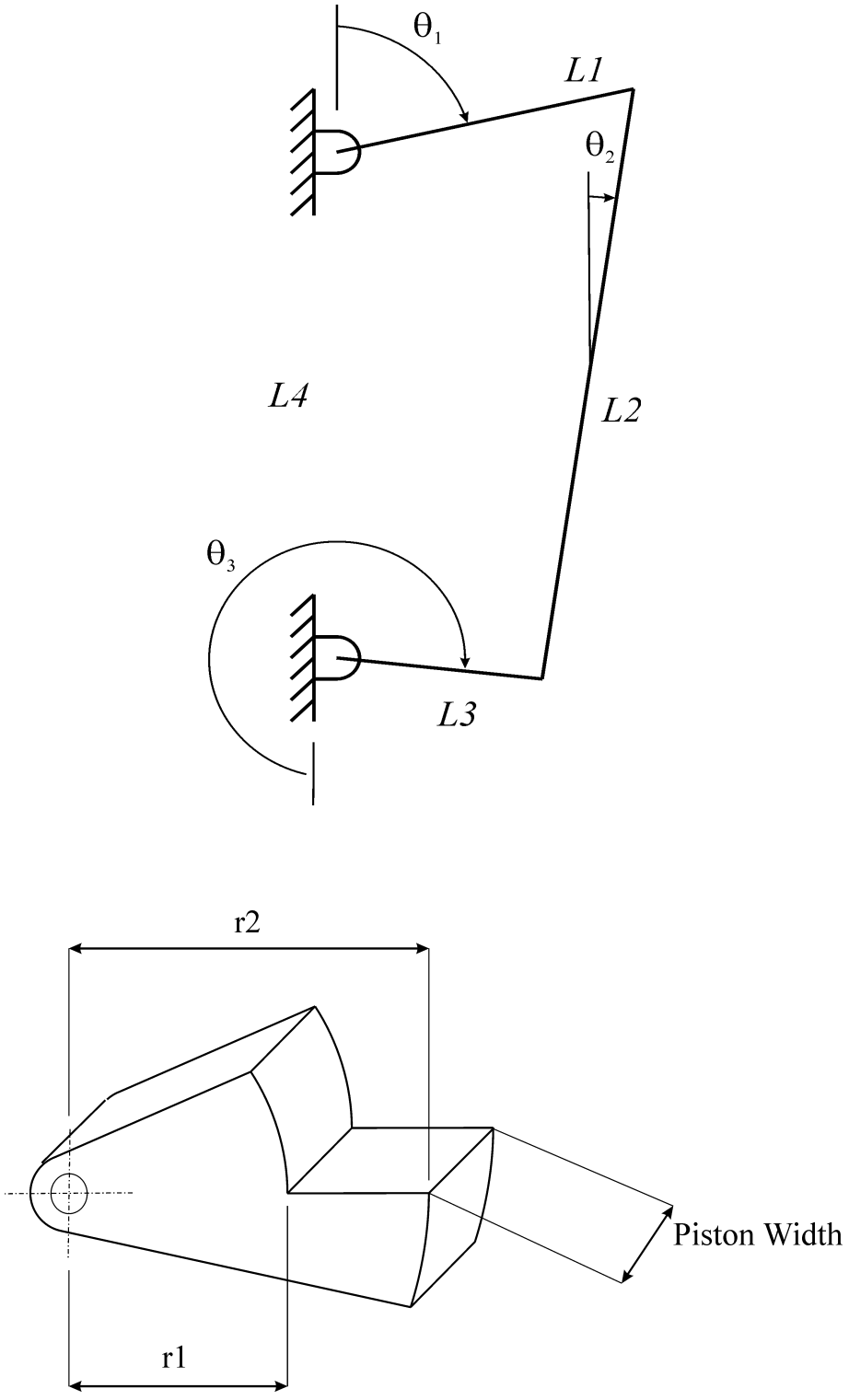


Figure 9-1. Pivotal four-bar kinematics diagram.

INPUT BOCK

```
!************************************************************
PivotalKinematics

direction  = 1
L1 = 0.034
L2 = 0.142
L3 = 0.11
L4 = 0.175
r1 = 0.074
r2 = 0.146
PistonWidth = 0.1
Vcl_cyl = 0.000047
Vcl_cc = 0.001
Vcl_bc = 0.00005

!************************************************************
```

VARIABLES

Direction: Direction of rotation of the crank as viewed in the figure. '1' for clockwise and '-1' for counter clockwise.

L1, L2, L3, L4, r1, r2, PistonWidth: Dimensions in metres specified by the figure.

Vcl_cyl: Cylinder clearance volume (m^3).

Vcl_cc: Crankcase clearance volume (m^3).

Vcl_bc: Pivotal boost chamber volume (m^3).

Kinematics

- Defines the kinematics for conventional slider crank engine simulations.

INPUT BOCK

```
!************************************************************
Kinematics

CR_cyl = 10
CR_cc = 1.5
bore = 0.086
stroke = 0.086
conrod = 0.18
```

VARIABLES

CR_cyl: Cylinder compression ratio

CR_cc: Crankcase compression ratio

Bore: Cylinder bore.

Stroke: Cylinder stroke.

Conrod: Connecting rod length.

Fuel

States the properties of the fuel and variables for the combustion model.

DIAGRAM

INPUT BOCK

```
!************************************************************
Fuel

ignition = 3.1
dw_burn = 0.52
Wiebe_a = 5.0
Wiebe_b = 2.0
FuelFile = f:\configtest\fuel.dat
lambda = 0.95
eta = 0.98
Tstar = 400

!************************************************************
```

VARIABLES

Ignition: Onset of ignition ABDC (rads)

dw_burn: Duration of burn (rads)

Wiebe_a: Wiebe combustion model coefficent.

Wiebe_b: Wiebe combustion model coefficent.

FuelFile: Specifies full path for the location of the fuel properties file.

Lambda: Fuel/Air equivalence ratio.

Eta: Combustion efficiency (0-1).

Tstar: Mean temperature (K) at which to calculate the heat capacity of species in all pipe elements.

Ambients

- Defines an ambient state for engine inlets and outlets.
- Separate ambient required for each inlet and outlet.

DIAGRAM

INPUT BOCK

```
!************************************************************
Ambients
2

ID      P       T       equiv   RH      Name
100     101325 300      1D12    0.3     InletAmbient
129     101325 300      1D12    0.3     OuletAmbient

!************************************************************
```

VARIABLES

First variable after 'Ambients' is an integer value equal to the number of ambient input rows to read.

ID: Integer identifier for the ambient

P: Ambient pressure (Pa)

T: Ambient temperature (K).

equiv: Air/Fuel equivalence ratio for the ambient.

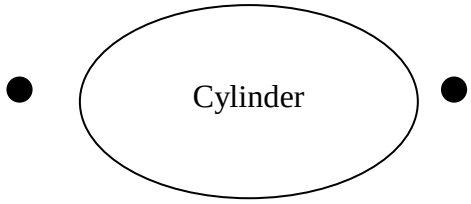
RH: Relative humidity of the ambient (0-1).

Name: Name of the ambient.

Cylinder

- A separate cylinder input block is required for each cylinder in the simulation.

DIAGRAM



INPUT BOCK

```
!************************************************************
Cylinder

name = cylinder
output = .true.
NumExhaustValves = 1
NumInletValves = 3
ExhaustValveID = 118
InletValveID = 113 115 117
WallTemp = 450
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 800
EffectiveDia = 0.08
Thead = 400
Tbore = 400
Tpiston = 400
HeadArea = 0
PistonArea = 0
```

VARIABLES

name: Name of cylinder element used for data output.

Output: Either '.true.' or '.false.' to specify if output is to be generated for the element.

NumExhaustValves: Number of exhaust valves the cylinder has.

NumInletValves: Number of inlet valves the cylinder has.

ExhaustValveID: The ID for each exhaust valve of the cylinder, e.g. if an engine has three exhaust valves, 118, 119 and 120, then 'ExhaustValveID = 118 1189 120'.

InletValveID: Similar to 'ExhaustValveID'.

Phase: Phase of the cylinder in the engine simulation ABDC (rads) from cylinder one.

InitialPressure (Pa), InitialTemperature (K): Initial conditions of the cylinder.

EffectiveDia: Effective cylinder diameter (m). Used for heat transfer coefficient calculations.

Thead: Mean head surface temperature (K). Input for heat transfer model.

Tbore: Mean bore surface temperature (K). Input for heat transfer model.

Tpiston: Mean piston surface temperature (K). Input for heat transfer model.

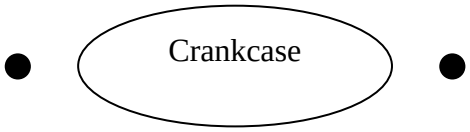
HeadArea: Surface area on the head (m^3) available for heat transfer.

PistonArea: Surface area on the piston (m^3) available for heat transfer.

CrankCase

- A separate input block is required for each crankcase element.

DIAGRAM



INPUT BOCK

```
!************************************************************
CrankCase

Type = 1
NumExhaustPorts = 3
NumInletPorts = 2
name = CrankCase
ExhaustPortID = 110 114 116
InletPortID = 105 109
WallTemp = 450
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 350
InitialLambda = 1D12
output = .true.

!************************************************************
```

VARIABLES

Type: Specifies the type of crankcase, '1' for a conventional crankcase and '2' for a pivotal boost chamber crankcase. Option '2' can only be used with pivotal kinematics.

NumExhaustPorts: Number of exhaust ports the crankcase has.

NumInletPorts: Number of inlet ports the crankcase has.

Name: Name of the crankcase as used in the output file.

ExhaustPortID: The ID for each exhaust port of the crankcase, e.g. if a crankcase has three exhaust ports, 118, 119 and 120, then 'ExhaustPortID = 118 1189 120'.

InletPortID: Similar to 'ExhaustPortID'.

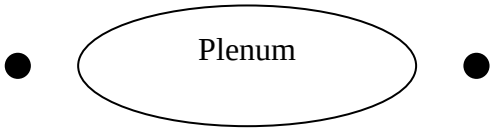
Phase: Phase of the crankcase in the engine simulation ABDC (rads) from crankcase one.

InitialPressure (Pa): InitialTemperature (K):, InitialLambda(A/F equivalence) Initial conditions of the crankcase.

Output: Either '.true.' or '.false.' to specify if output is to be generated for the element.

Plenum

DIAGRAM



INPUT BOCK

```
!************************************************************
Plenum

name = ReedPlenum
NumExhaustPorts = 2
NumInletPorts = 1
ExhaustPortID = 102 104
InletPortID = 101
InitialPressure = 1E5
InitialTemperature = 300
InitialLambda = 1D12
Volume = 0.0005
output = .true.

!************************************************************
```

VARIABLES

Name: Name of the plenum as used in the output file.

NumExhaustPorts: Number of exhaust ports the plenum has.

NumInletPorts: Number of inlet ports the plenum has.

ExhaustPortID: The ID for each exhaust port of the plenum, e.g. if the plenum has three exhaust ports, 118, 119 and 120, then 'ExhaustPortID = 118 1189 120'.

InletPortID: Similar to 'ExhaustPortID'.

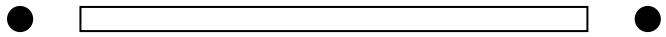
InitialPressure (Pa):, InitialTemperature (K):, InitialLambda: Initial conditions of the plenum.

Volume: Volume of the plenum (m_3).

Output: Either '.true.' or '.false.' to specify if output is to be generated for the element.

Pipes

DIAGRAM



INPUT BOCK **(Also see the generic Pivotal engine example in this chapter)**

```
!************************************************************
Pipes
2
```

N	InID	OutID	InDia	OutDia	len	Pint	Vint	Tint	Lam	BF	output
	name	fuelInjection		position							
#nodes	103	106	#dia1	#dia1	0.2	101325	0	300	1D12	0	.true.
	BoostPipe		.false.	0							
#nodes	111	112	#dia1	#dia1	0.1	101325	0	300	1	0	.true.
	TrsfrPipe	.true.	0.05								

```
!************************************************************
```

VARIABLES

First variable after pipes specifies the number of pipe elements to read in.

N: Number of nodes (faces) in the pipe discretisation.

InID: Integer ID of the pipe inlet.

OutID: Integer ID of the pipe outlet.

InDia (m): Diameter at the pipe inlet

OutDia (m): Diameter at the pipe outlet.

len (m): length of the pipe.

Pint (Pa): Initial pipe pressure.

Vint (m/s): Initial pipe velocity.

Tint (K): Intial pipe temperature.

Lam: Air/Fuel equivalence ratio initial condition for the pipe.

BF: Burnt fraction initial condition for the pipe.

Output: Specify '.true.' to generate output for the pipe otherwise set to '.false.'.

Name: Name of the pipe. Used in output file.

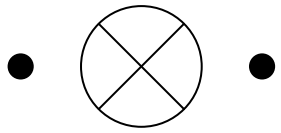
FuelInjection: '.true.' if the pipe has a fuel injector otherwise set to '.false.'.

Position (m): Position of fuel injector from the front of the pipe.

LiftValves

- Virtual element that connect thermodynamic volume elements to pipe elements.

DIAGRAM



INPUT BOCK

```
!************************************************************
LiftValves
2

CylID   PipeID  AreaID  Type    output  Mult    Name
102     103     201     exhaust .true.  1       BoostPipeReed
107     106     203     inlet   .true.  1       BoostPipeOutlet

!************************************************************
```

VARIABLES

First variable after LiftValves determines the number of input lines the program will read.

CylID: Integer of the cylinder, plenum, ambient or crankcase node to which the valve is connected.

PipeID: Integer identifier of the pipe ID to which the valve is connected.

AreaID: Integer ID of the throat area element.

Type: 'Exhaust' or 'Inlet' valve. Exhaust connects a thermodynamic volume to a pipe inlet. Inlet connects a pipe outlet to a thermodynamic volume.

Output: '.true.' or '.false' to determine if output is generated for the element.

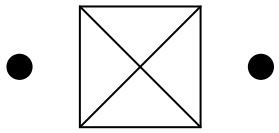
Mult: Flux multiplier for the cylinder flux, but not the pipe flux. Allows multiple identical pipes with out including each identical pipe element.

Name: Name of the valve element used in output file.

OrificeValves

- Virtual element that connects any two thermodynamic elements.

DIAGRAM



INPUT BOCK

```
!************************************************************
OrificeValves
5

ExID    InID    AreaID  FluxMult        output  name
100     101     200     1               .ture.  throtle
104     105     202     1               .true.  CCReedPort
108     109     204     2               .true.  BoostTrsfrPort
116     117     208     1               .true.  LeakPort
114     115     207     2               .true.  AuxTrsfrOrifice

!************************************************************
```

VARIABLES

ExID: ID of the exhaust to which the valve is attached.

InID: ID of the inlet to which the valve is attached.

AreaID: Integer ID of the throat area element.

FluxMulti: Multiplier for the valve flux. Used for multiple identical valves.

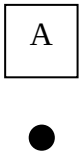
Output: Specify '.true.' to generate output for the pipe otherwise set to '.false.'.

Name: Name of the valve. Used in output file.

LiftCurves

- Specifies a throat area that can vary with crank angle.

DIAGRAM



INPUT BOCK

```
!************************************************************
LiftCurves
4

ID      N       vo      dur     area    name    DataFile
206     20      315     0.5     1.0     TrsfrPipe       f:\configtest\expro1.dat
207     20      315     0.5     0.5     trsfrOrifice    f:\configtest\expro1.dat
204     20      90      1       1.0     boosttrsfr      f:\configtest\expro1.dat
209     20      270     1       1.7     exhaust         f:\configtest\expro1.dat

!************************************************************
```

VARIABLES

ID: ID of the area element.

N: Number of data points used to specify the area curve.

vo (deg): Crank angle at which the valve opens (ABDC).

Dur: Multiplier for the area curve duration.

Area: Multiplier for the area curve throat area.

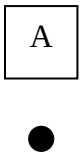
Name: Name of the area element.

DataFile: Full path of the data file that contains the crank angle versus throat area curve.

FixedAreas

- Input throat areas that are constant with crank angle.

DIAGRAM



INPUT BOCK

```
!************************************************************
FixedAreas
5

ID      Dia     Name
200     0.05    Throttle
203     #dia1   BoostIntakeArea

!************************************************************
```

VARIABLES

Value after ‘FixedAreas’ specifies how many fixed area elements the program will read.

ID: Integer identifier.

Dia (m): Diameter of the area element.

Name: Name of the area element.

Reed

- Input block for reed throat area elements
- Area element that is a function of the pressure in the connecteing pipe/volume elements.
- Separate input block required for each reed.

DIAGRAM

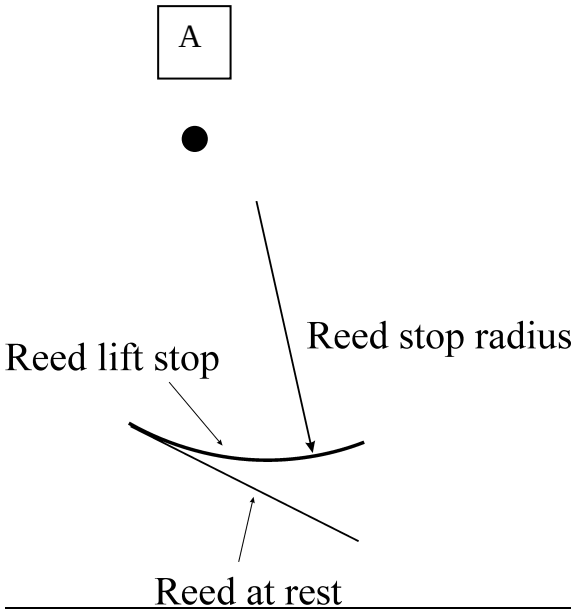


Figure 9-2. Reed lift stop.

INPUT BOCK

```
!************************************************************
Reed

name = CCReed
ID = 202
NumNodes = 6
ReedWidth = 0.015
ReedThickness = 0.0002
ReedLength = 0.033
BetaOne = 0.5
BetaTwo = 0.5
Modulus = 210e9
Density = 7800
NumPetals = 6
ReedAngle = 30
ReedStopRadius = 0.06

!************************************************************
```

VARIABLES

Name: Name of the reed.

NumNodes: Number of nodes in the finite element discretisation of the reed.

ReedWidth (m): Reed width

ReedThickness: (m) Reed thickness

ReedLength: (m) Reed length

Modulus (Pa): Young's modulus of the reed material.

Density (kg/m^3): Density of the reed material

NumPetals: Number of reed petals

ReedAngle (deg): Angle of the reed at rest to the mean flow direction.

ReedStopRadius (m): Radius of the reed lift stop radius (see Figure 9-2).

Lift Curve File

The lift curve file is an ASCII text file containing two columns of data that specify a curve of effective throat area against crank angle. The first column gives the crank ankle in degrees, starting at zero when the valve first opens. The second column contains the effective throat area (m^3).

LIFT CURVE TEXT FILE EXAMPLE.

0.0000000e+000	0.0000000e+000
9.4736800e+000	1.2255851e-005
1.8947370e+001	4.7695293e-005
2.8421050e+001	1.0247791e-004
3.7894740e+001	1.7066716e-004
4.7368420e+001	2.4487368e-004
5.6842110e+001	3.1705604e-004
6.6315790e+001	3.7939215e-004
7.5789470e+001	4.2512695e-004
8.5263160e+001	4.4930434e-004
9.4736840e+001	4.4930434e-004
1.0421053e+002	4.2512695e-004
1.1368421e+002	3.7939215e-004
1.2315789e+002	3.1705604e-004
1.3263158e+002	2.4487368e-004
1.4210526e+002	1.7066716e-004
1.5157895e+002	1.0247791e-004
1.6105263e+002	4.7695293e-005
1.7052632e+002	1.2255851e-005
1.8000000e+002	0.0000000e+000

Fuel File

This file contains properties of the fuel being used

FUEL FILE EXAMPLE:

****

A1	-2.3143
A2	759.8981
A3	-409.1408
A4	85.3620
A5	-0.1295
A6	-254.1822
a	8
b	18
c	0
d	0
TFuel	3.5d2
Cfuel	2.1e3
Hvap	308e3

****

VARIABLES

$$e_{fuel} = A_1 t_r + A_2 t_r^{\,2} + A_3 t_r^{\,3} + A_4 t_r^{\,4} - \frac{A_5}{t_r} + A_6$$

$$t_r = \frac{T(K)}{1000}$$

$\mathbf{e}_{\text{fuel}}$ (kJ/kg/K)

A_1: Fuel specific energy polynomial coefficient

A_2: Fuel specific energy polynomial coefficient

A_3: Fuel specific energy polynomial coefficient

A_4: Fuel specific energy polynomial coefficient

A_5: Fuel specific energy polynomial coefficient

A_6: Fuel specific energy polynomial coefficient

a: mean number of carbon atoms per fuel molecule.

b: mean number of hydrogen atoms per fuel molecule.

c: mean number of oxygen atoms per fuel molecule.

d: mean number of nitrogen atoms per fuel molecule.

Tfuel (K): Temperature of the liquid fuel.

Cfuel (J/kg/K): Specific heat capacity of liquid fuel.

Hvap (J/kg/K): Latent heat of vaporisation of the liquid fuel.

Output files

The simulation creates three types of output files.

c:\inputpath\inputfilename.i

Information output text file that records any messages output during the simulation.

c:\inputpath\inputfilename*X*.out.

A detailed Matlab output file for each parameter case, where *X* is the parameter case number e.g. c:\inputpath\inputfilename1.out. The directory path is the same as for the input file.

These files are loaded into the Matlab environment by typing

Load c:\inputpath\inputfile*X*.out at the Matlab command prompt.

c:\inputpath\inputfile.sum.

Summary text file with engine speed and indicated engine power output

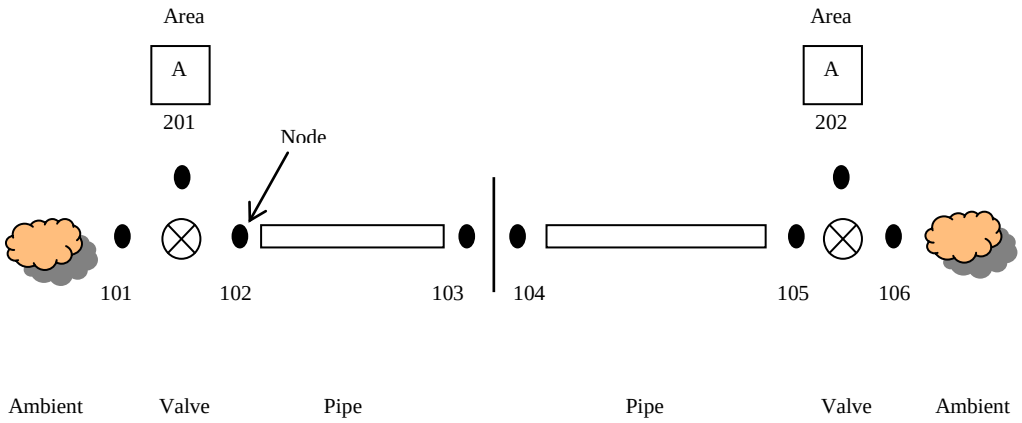
The generation of Matlab files for output is taken care of by the Matlab application program interface (API). The engine simulation uses the Matlab dynamic link libraries fmx.dll, fmat.dll, libmx.dll and libut.dll. These dynamic link libraries must be located in the same directory as the engine simulation program and are supplied with Matlab. More information regarding the Matlab API can be found in the Matlab help.

Examples

A simple shock tube flow example

The shock tube problem models the rupture of a diaphragm in a pipe with different initial pressure and density on either side of the diaphragm. This is modelled by two pipe elements with separate initial conditions in each pipe. Valve elements are used to model the pipe boundary conditions, the areas A_{201} and A_{202} are set to the diameter of the pipe.

DIAGRAM



INITIAL CONDITIONS

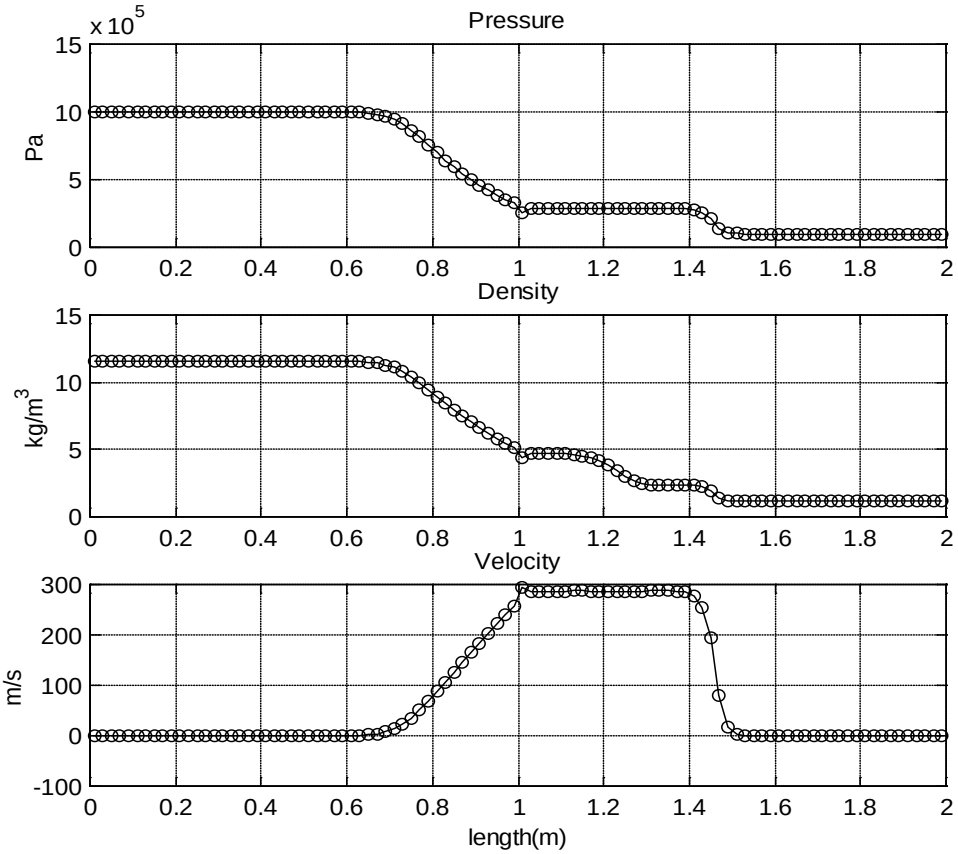
Temperature:	300 K
Pipe diameter:	0.01 m
Pipe length:	1m (each)
Left side pressure:	10 bar
Right side pressure:	1 bar
Nodes:	51 (each pipe)

INPUT FILE

```
!************************************************************
ParameterMatrix
7
1

#rpm    5000
#dia    0.01
#len    1
#Phigh  1D6
#Plow   1D5
#T      300
#nodes 51

!************************************************************
Setup

CFL = 0.4
soln_tol = 1D-4
MaxCycles = 1
spatial_acu = 2
MaxSimTime = 30
KinematicsType = none
NumStrokes = 2

!************************************************************
Junctions
1

UpID    DownID Dia      name
103     104    #dia     Junc

!************************************************************
Fuel

ignition = 3.1
dw_burn = 0.52
Wiebe_a = 5.0
Wiebe_b = 2.0
FuelFile = f:\thesis\examples\GenericPivotalEngine\fuel.dat
lambda = 0.95
eta = 0.98
Tstar = 400

!************************************************************
Ambients
2

ID      P       T       equiv   RH      Name
101     #Phigh  #T      1D12    0.3     InletAmbient
106     #Plow   #T      1D12    0.3     OuletAmbient
```

```
!************************************************************
Pipes
2

N       InID    OutID   InDia   OutDia  len     Pint    Vint    Tint    Lam     BF      output
        name    fuelInjection   position
#nodes  102     103     #dia    #dia    #len    #Phigh  0       #T      1D12    0       .true.
        LeftPipe .false. 0
#nodes  104     105     #dia    #dia    #len    #Plow   0       #T      1D12    0       .true.
        RightPipe.false. 0.05

!************************************************************
LiftValves
2

CylID   PipeID  AreaID  Type    output  Mult    Name
101     102     201     exhaust.true.   1       InletValve
106     105     202     inlet   .true.  1       OutletValve

!************************************************************
FixedAreas
2

ID      Dia     Name
201     #dia    InletArea
202     #dia    OutletArea
```

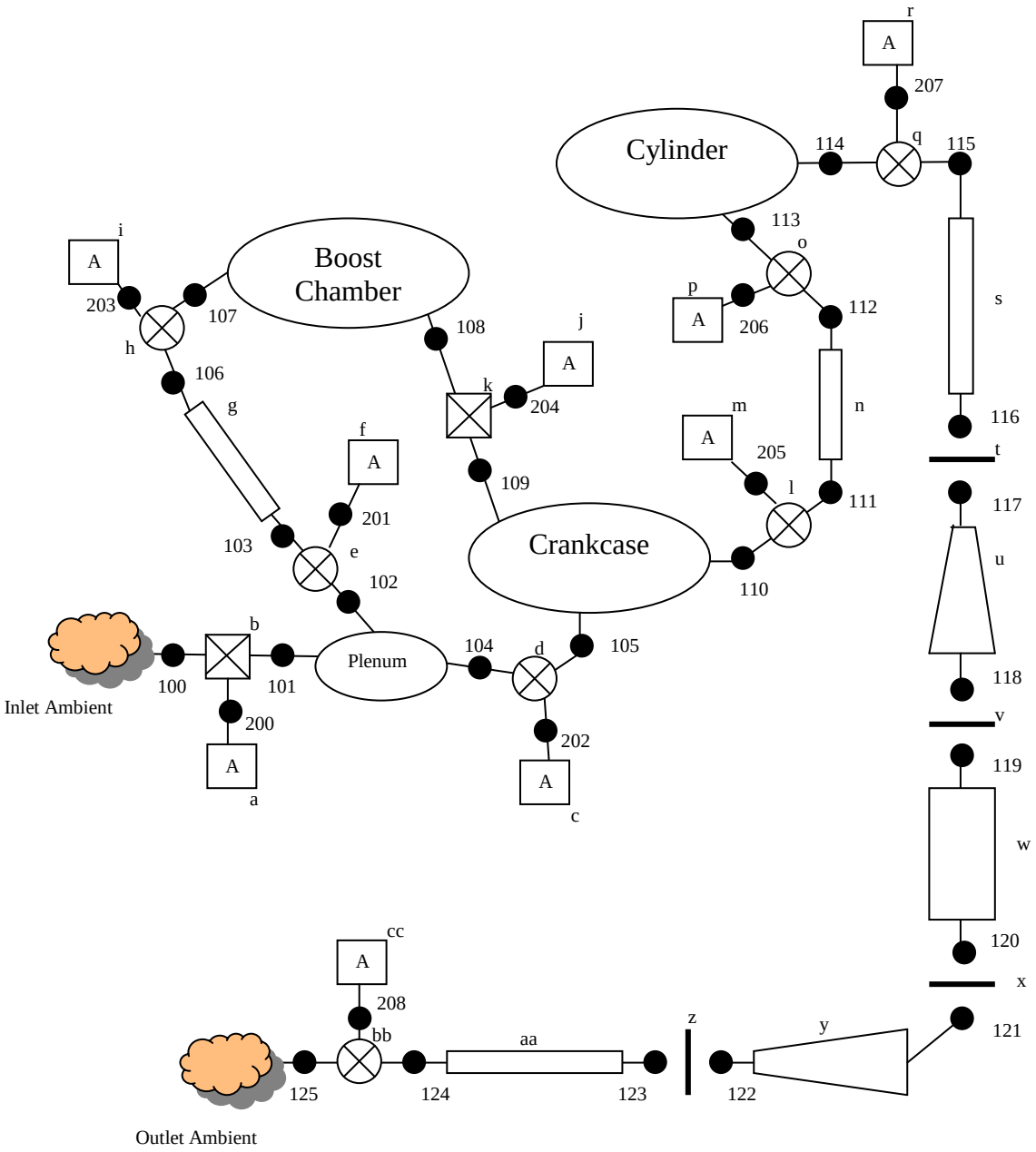
OUTPUT


Pressure
x 10^5
Pa
15
10
5
0
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
Density
kg/m^3
15
10
5
0
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
Velocity
m/s
300
200
100
0
-100
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2
length(m)

Generic pivotal engine example

The following example illustrates how to model a complete engine. Every modelling element has been used in this example.

DIAGRAM



LEGEND

a	ThrottleArea
b	Throttle
c	CrankCaseOrificeReed
d	CrankCaseOrifice
e	BoostReedValve
f	BoostReedArea
g	BoostPipe
h	BoostPipeOutlet
i	BoostPipeOutletArea
j	BoostTransferArea
k	BoostTransferOrifice
l	TransferPipeInlet
m	TransferPipeInletArea
n	TransferPipe
o	InletValve
p	InletValveArea
q	ExhaustValve
r	ExhaustValveArea
s	ExhaustPipe1
t	Junction 1
u	ExhaustPipe2
v	Junction 2
w	ExhaustPipe3
x	Junction 3
y	ExhaustPipe4
z	Junction4
aa	ExhaustPipe5
bb	EngineOutlet
cc	EngineOutletArea

INPUT FILE

```
!************************************************************
ParameterMatrix
5
12

#rpm    1000    1500    2000    2500    3000    3500    4000    4500    5000    5500    6000
        6500
#dia1   0.05    0.05    0.05    0.05    0.05    0.05    0.05    0.05    0.05    0.05    0.05
        0.05
#dia2   0.1     0.1     0.1     0.1     0.1     0.1     0.1     0.1     0.1     0.1     0.1
        0.1
#nodes  11      11      11      11      11      11      11      11      11      11      11
        11
#maxc   50      50      50      50      50      50      50      50      50      50      50
        50

!************************************************************
Setup

CFL = 0.6
soln_tol = 1D-4
MaxCycles = #maxc
spatial_acu = 2
MaxSimTime = 30
KinematicsType = FB
NumStrokes = 2

!************************************************************
Junctions
4

UpID    DownID  Dia     name
116     117     #dia1   Junction1
118     119     #dia2   Junction2
120     121     #dia2   Junction3
122     123     #dia1   Junction4

!************************************************************
PivotalKinematics

direction  = 1
L1 = 0.034
L2 = 0.142
L3 = 0.11
L4 = 0.175
r1 = 0.074
r2 = 0.146
PistonWidth = 0.1
Vcl_cyl = 0.000047
Vcl_cc = 0.001
Vcl_bc = 0.00005

!************************************************************
Fuel

ignition = 3.1                  % Onset of combustion in Weibe model.

dw_burn = 0.52                  % Weibe function input.
Wiebe_a = 5.0                   % Weibe curve parameter.
Wiebe_b = 2.0
```

```
FuelFile = f:\thesis\examples\GenericPivotalEngine\fuel.dat
lambda = 0.95
eta = 0.98
Tstar = 400
```

```
!************************************************************
Ambients
2

ID      P       T       equiv   RH      Name
100     101325 300      1D12    0.3     InletAmbient
125     101325 300      1D12    0.3     OuletAmbient

!************************************************************
Cylinder

name = cylinder
output = .true.
NumExhaustValves = 1
NumInletValves = 1
ExhaustValveID = 114
InletValveID = 113
WallTemp = 450
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 800
EffectiveDia = 0.08
Thead = 400
Tbore = 400
Tpiston = 400
HeadArea = 0
PistonArea = 0

!************************************************************
CrankCase

Type = 1
NumExhaustPorts = 1
NumInletPorts = 2
name = CrankCase
ExhaustPortID = 110
InletPortID = 105 109
WallTemp = 450
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 350
InitialLambda = 1D12
output = .true.

!************************************************************
CrankCase

Type = 2
NumExhaustPorts = 1
NumInletPorts = 1
name = BoostChamber
ExhaustPortID = 108
InletPortID = 107
WallTemp = 450
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 350
InitialLambda = 1D12
output = .true.

!************************************************************
Plenum
```

name = Plenum
NumExhaustPorts = 2
NumInletPorts = 1
ExhaustPortID = 102 104
InletPortID = 101
WallTemp = 300
InitialPressure = 1E5
InitialTemperature = 300
InitialLambda = 1D12
Volume = 0.0005
output = .true.

!**
Pipes
7

N	InID	OutID	InDia	OutDia	len	Pint	Vint	Tint	Lam	BF	output
	name	fuelInjection		position							
#nodes	103	106	#dia1	#dia1	0.2	101325	0	300	1D12	0	.true.
	BoostPipe		.false.	0							
#nodes	111	112	#dia1	#dia1	0.1	101325	0	300	1	0	.true.
	TransferPipe		.true.	0.05							
#nodes	115	116	#dia1	#dia1	0.5	101325	0	450	1	1	.true.
	ExhaustPipe1		.false.	0							
#nodes	117	118	#dia1	#dia2	0.5	101325	0	450	1	1	.true.
	ExhaustPipe2		.false.	0							
#nodes	119	120	#dia2	#dia2	0.5	101325	0	450	1	1	.true.
	ExhaustPipe3		.false.	0							
#nodes	121	122	#dia2	#dia1	0.5	101325	0	450	1	1	.true.
	ExhaustPipe4		.false.	0							
#nodes	123	124	#dia1	#dia1	0.5	101325	0	450	1	1	.true.
	ExhaustPipe5		.false.	0							

!**
LiftValves
6

CylID	PipeID	AreaID	Type	output	Mult	Name
102	103	201	exhaust	.true.	1	BoostReedValve
107	106	203	inlet	.true.	1	BoostPipeOutlet
110	111	205	exhaust	.true.	2	TransferPipeInlet
113	112	206	inlet	.true.	2	InletValve
114	115	209	exhaust	.true.	1	ExhaustValve
125	124	210	inlet	.true.	1	EngineOutlet

!**
OrificeValves
3

ExID	InID	AreaID	FluxMult	output	name
100	101	200	1	.ture.	throtle
104	105	202	1	.true.	CrankCaseOrifice
108	109	204	2	.true.	BoostTransferOrifice

!**
LiftCurves
3

ID	N	vo	dur	area	name	DataFile
206	20	315	0.5	1.0	InletValveArea	
	f:\thesis\examples\GenericPivotalEngine\AreaProfile.dat					
204	20	90	1	1.0	boosttransferArea	
	f:\thesis\examples\GenericPivotalEngine\AreaProfile.dat					
207	20	270	1	1.7	ExhaustValveArea	
	f:\thesis\examples\GenericPivotalEngine\AreaProfile.dat					

```
!************************************************************
FixedAreas
4

ID      Dia     Name
200     0.05    Throttle
203     #dia1   BoostPipeOutletArea
205     #dia1   TrsfrPipeInletArea
2080    #dia1   OutletArea

!************************************************************
Reed

name = CrankCaseOrificeReed
ID = 202
NumNodes = 6
ReedWidth = 0.015
ReedThickness = 0.0002
ReedLength = 0.033
BetaOne = 0.5
BetaTwo = 0.5
Modulus = 210e9
Density = 7800
NumPetals = 6
ReedAngle = 30
ReedStopRadius = 0.06

!************************************************************
Reed

name = BoostReedArea
ID = 201
NumNodes = 6
ReedWidth = 0.015
ReedThickness = 0.0002
ReedLength = 0.033
BetaOne = 0.5
BetaTwo = 0.5
Modulus = 210e9
Density = 7800
NumPetals = 3
ReedAngle = 30
ReedStopRadius = 0.06
```

OUTPUT

Figures 9-3 to 9-8 give a graphical overview of the main output from a simulation run. The graphical output in each graph, except for Figure 9-4, is for parameter case, 6, which is at 3500rpm engine speed. Figure 9-4 gives a summary of the indicated engine output for all parameter cases that in this case is for the speed range from 1000 rpm to 6500 rpm.

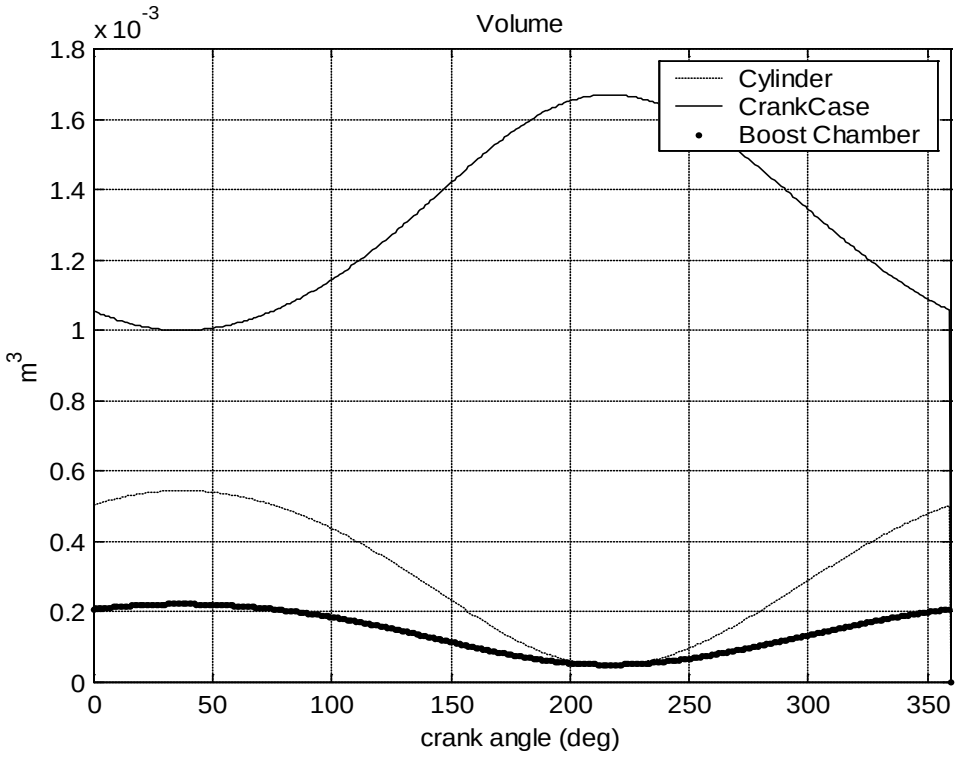


Figure 9-3. Output of the cylinder, crankcase and Pivotal boost chamber volume through one revolution of the crankshaft.

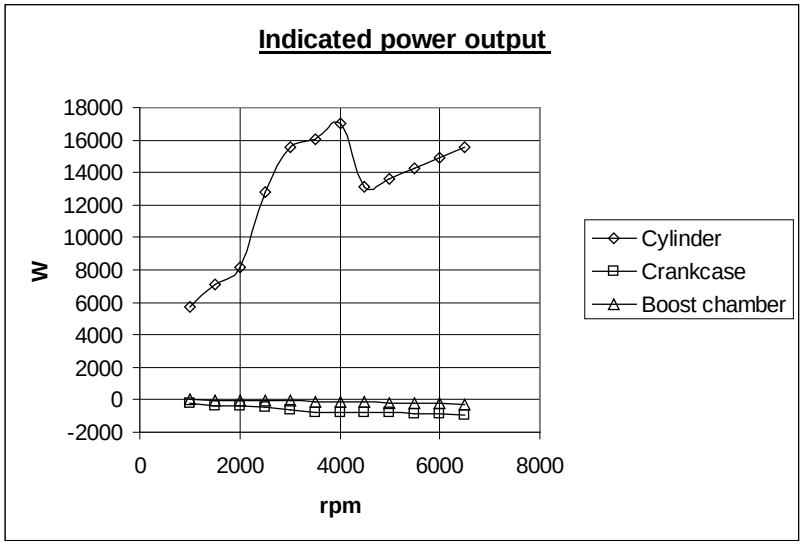


Figure 9-4. Indicate power output from engine.

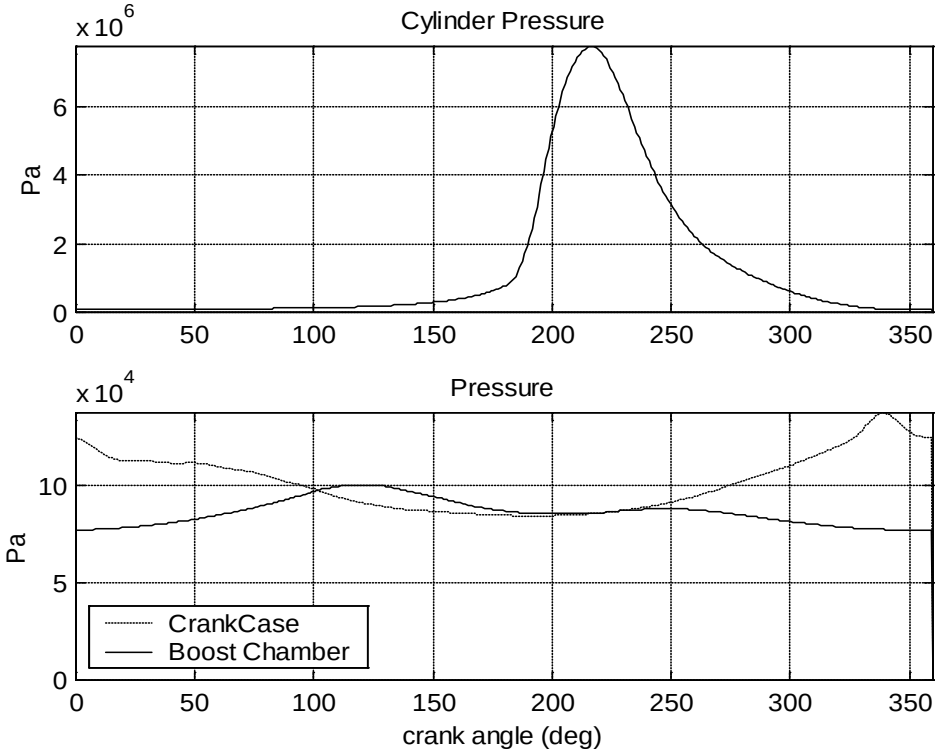


Figure 9-5. Variation of pressure in thermodynamic components of the engine.

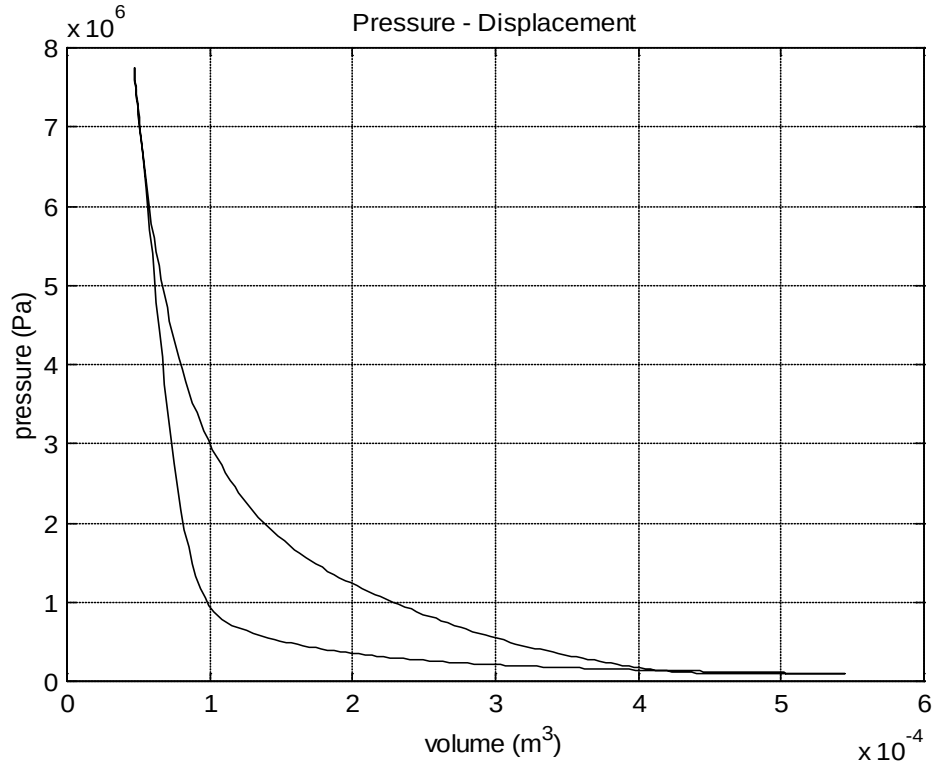


Figure 9-6. Pressure plotted against the volume of the pivotal "cylinder" volume.

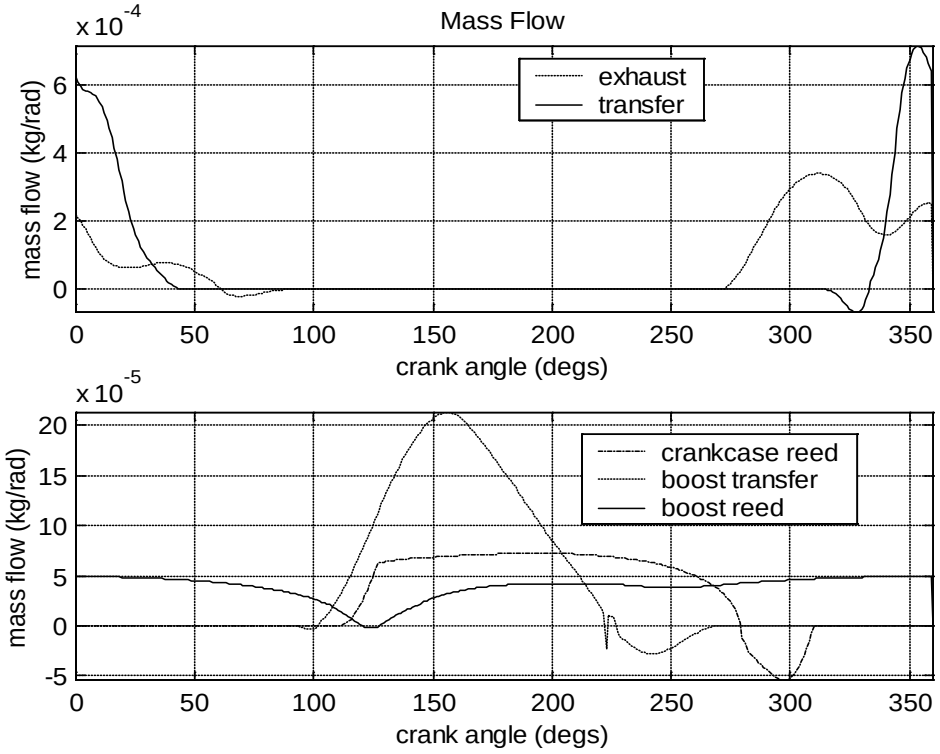


Figure 9-7. Port and valve mass flow rates. Note on the boost transfer mass flow the negative spike between 200 and 250 degrees. The spike is a result of a bug in the orifice model that could not be resolved. This spike only occurs as flow reverses through an orifice valve.

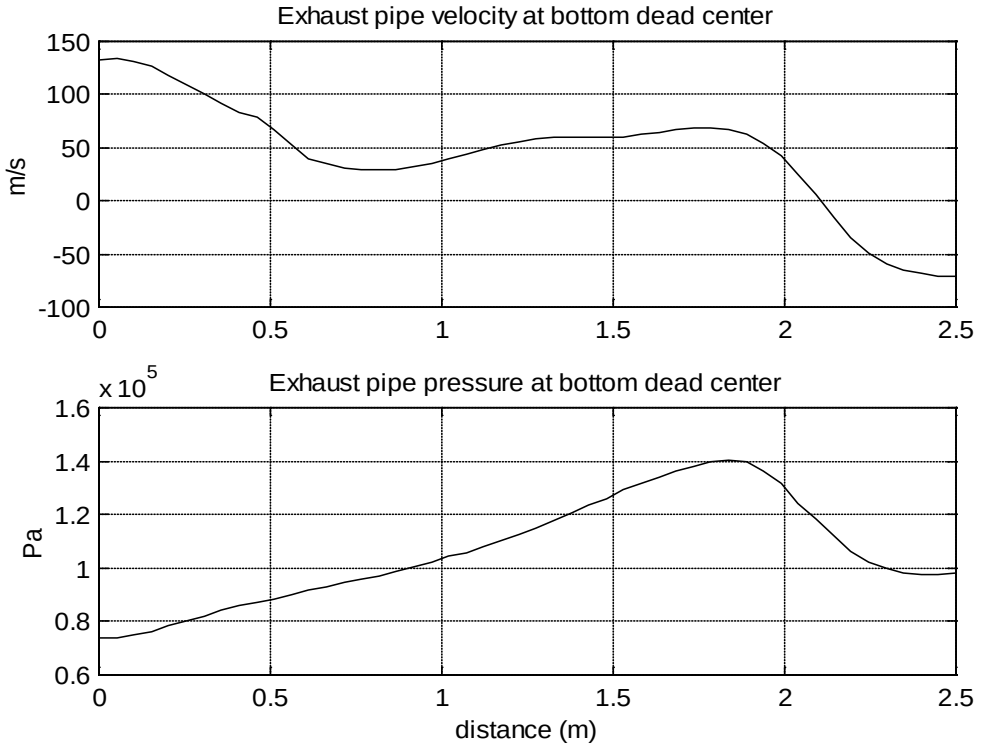
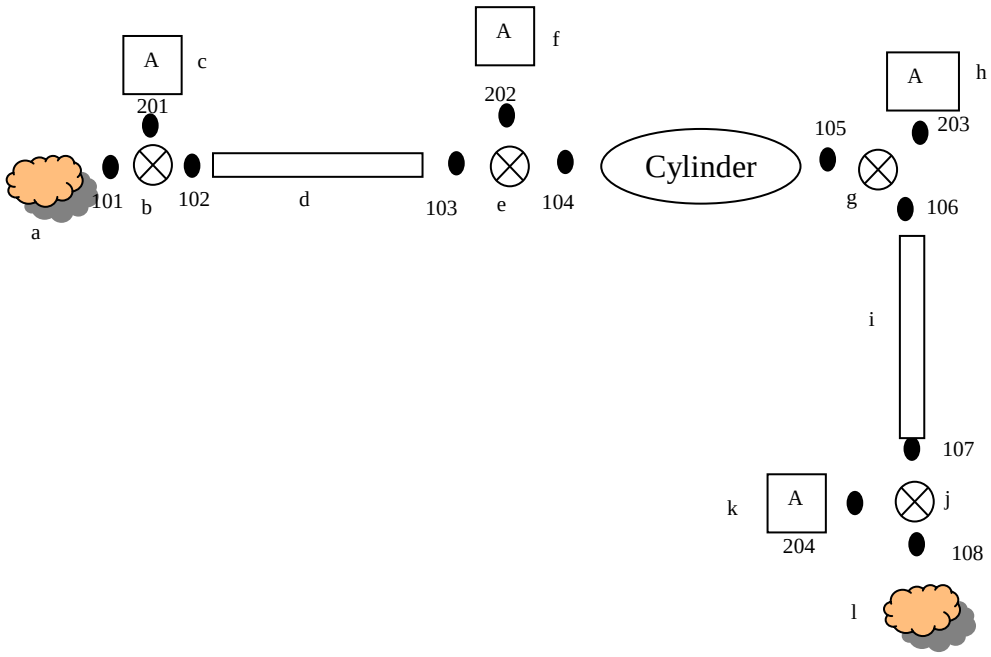


Figure 9-8. Pressure and velocity along the exhaust pipe at BDC

Single cylinder 4-stroke

This example models a simple single cylinder four-stroke engine. The inlet ambient is set up as an air fuel mixture so that no fuel injector is required.

DIAGRAM



LEGEND

a	InletAmbient
b	Valve1
c	InletArea
d	LeftPipe
e	Valve2
f	InletValveArea
g	Valve3
h	ExhaustValveArea
i	RightPipe
j	Valve4
k	OutletArea
l	OutletAmbient

INPUT FILE

```
!************************************************************
ParameterMatrix
3
10

#rpm    1500    2000    2500    3000    3500    4000    4500    5000    5500    6000
#dia    0.040   0.04    0.04    0.04    0.04    0.04    0.04    0.04    0.04    0.04
#len    0.4     0.4     0.4     0.4     0.4     0.4     0.4     0.4     0.4     0.4

!************************************************************
Setup

CFL = 0.4
soln_tol = 1D-4
MaxCycles = 12
spatial_acu = 2
MaxSimTime = 10
KinematicsType = SC
NumStrokes = 4

!************************************************************
Fuel

ignition = 2.9                          % Onset of combustion in Weibe model.

dw_burn = 0.45                          % Weibe function input.
Wiebe_a = 5.0                           % Weibe curve parameter.
Wiebe_b = 2.0
FuelFile =f:\thesis\examples\single4\fuel.dat
lambda = 0.9
eta = 1
Tstar = 400

!************************************************************
Kinematics

CR_cyl = 10
CR_cc = 1.5
bore = 0.08
stroke = 0.08
conrod = 0.14

!************************************************************
Ambients
2

ID      P       T       equiv   RH      Name
101     101325  300     0.9     0.3     InletAmbient
108     101325  300     1D12    0.3     OuletAmbient
```

```
!************************************************************
Cylinder

name = cylinder
output = .true.
NumExhaustValves = 1
NumInletValves = 1
ExhaustValveID = 105
InletValveID = 104
WallTemp = 52
Phase = 0
InitialPressure = 101325e+0
InitialTemperature = 800
EffectiveDia = 0.08
Thead = 520
Tbore = 520
Tpiston = 520
HeadArea = 0.005
PistonArea = 0.005

!************************************************************
Pipes
2

N       InID    OutID   InDia   OutDia  len     Pint    Vint    Tint    Lam     BF
        output  name    fuelInjection   position
21      102     103     #dia    #dia    #len    101325  0       300     1D12    0
        .true.  LeftPipe .false. 0
21      106     107     #dia    #dia    #len    101325  0       300     1D12    0
        .true.  RightPipe .false. 0.05

!************************************************************
LiftValves
4

CylID   PipeID  AreaID  Type    output  Mult    Name
101     102     201     exhaust .true.  1       Valve1
104     103     202     inlet   .true.  1       Valve2
105     106     203     exhaust .true.  1       Valve3
108     107     204     inlet   .true.  1       Valve4

!************************************************************
LiftCurves
2

ID      N       vo      dur     area    name    DataFile
202     18      510     1       0.6     InletValveArea  f:\thesis\examples\single4\inlet.dat
203     19      285     1       0.6     ExhaustValveArea
        f:\thesis\examples\single4\exhaust.dat

!************************************************************
FixedAreas
2

ID      Dia     Name
201     #dia    InletArea
204     #dia    OutletArea
```

OUTPUT

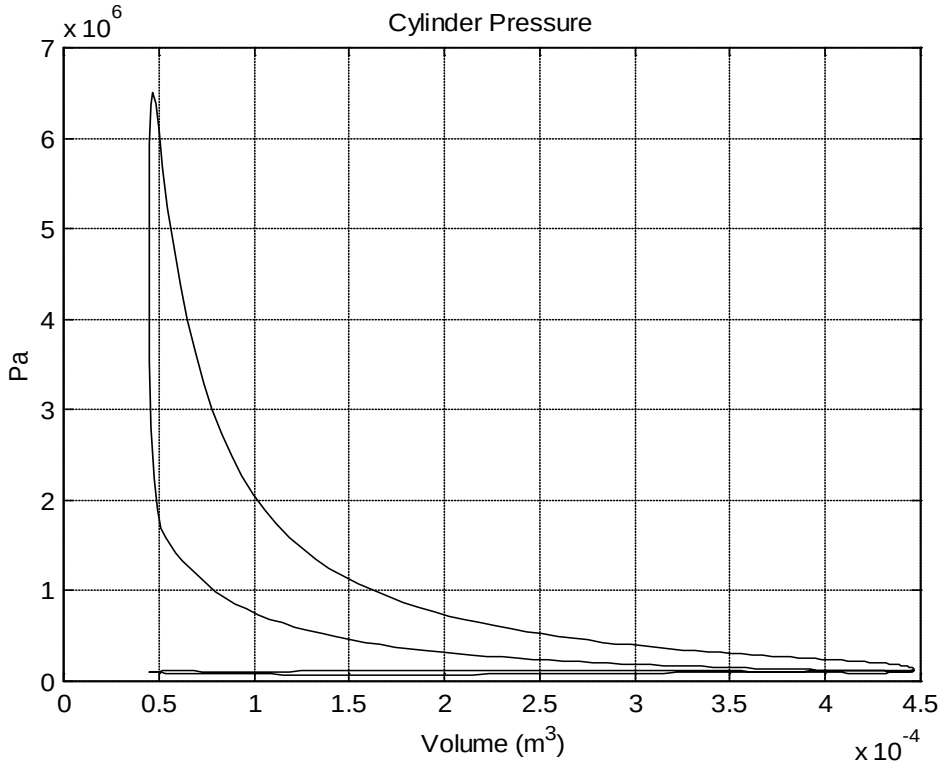


Figure 9-9. Pressure plotted against cylinder volume at 3000 rpm.

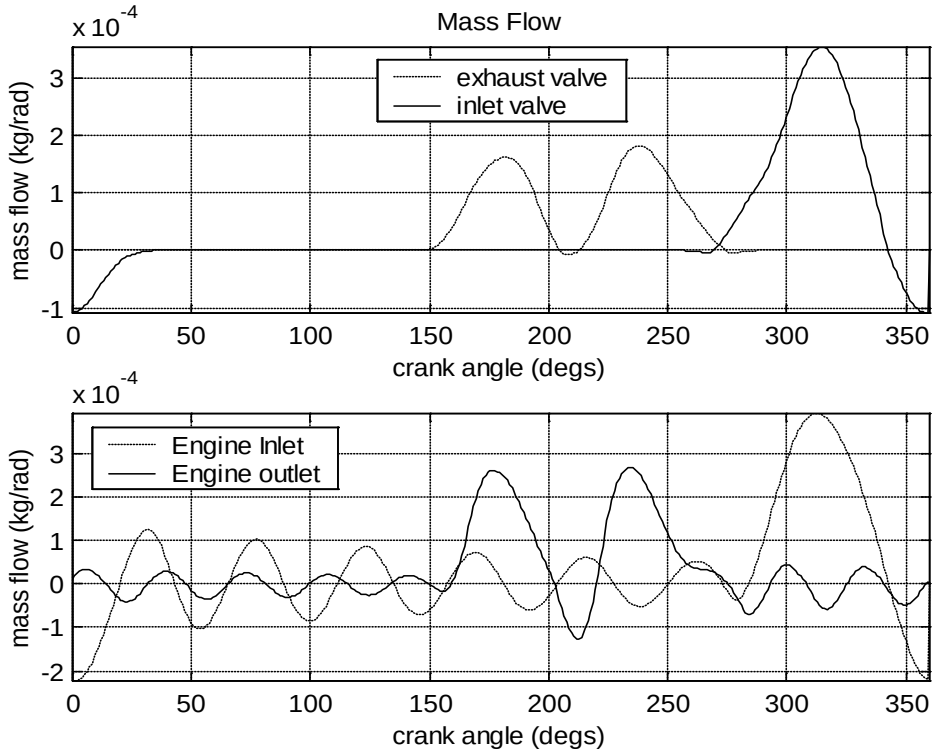


Figure 9-10. Port and valve mass flow rates at 3000 rpm.

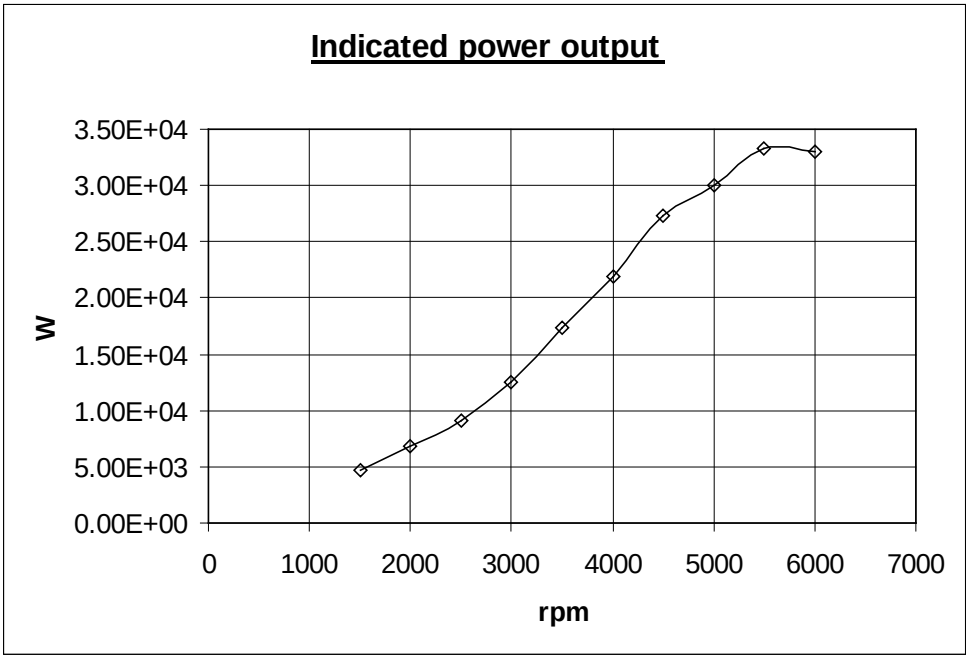


Figure 9-11. Performance of the simple, single cylinder 4-stroke engine.